

Ferramenta de Apoio ao Ensino Prático de Computação Ubíqua

Maria Joelma Pereira Peixoto Wellington W. F. Sarmiento Windson Viana
Universidade Federal do Ceará (UFC) Universidade Federal do Ceará (UFC) Universidade Federal do Ceará (UFC)
Fortaleza – CE – Brasil Fortaleza – CE – Brasil Fortaleza – CE – Brasil
joelmapereira@great.ufc.br wwagner@virtual.ufc.br windson@great.ufc.br

ABSTRACT

The term Ubiquitous Computing emerged in the late 80s, in which Mark Weiser, a scientist from XEROX PARC, imagined a future where technology would be so ingrained into the daily lives of people that they would not notice the apparatus involved. The computer would be embedded in numerous devices. These concepts began to be studied in various universities in courses related to Computer Science, which sought to develop the study of ubiquitous systems and technologies. In general, these courses have practical classes or even small ubiquitous application development projects. However, one of the greatest difficulties in teaching Ubiquitous computing is putting into practice the theoretical knowledge acquired, since the existing tools already requires strong programming skills or is not designed for educational purposes. As a result, this paper aims to design, develop and evaluate a tool that can help the practical classes of teachers who teach classes involving concepts of Ubiquitous Computing. Through the User-Centered Design (UCD), the system components are tested with real users in order to validate the proposed tool. The results of the first tests with users are analyzed and presented in this paper.

RESUMO

O termo Computação Ubíqua surgiu no final da década de 80, na qual Mark Weiser, cientista da XEROX PARC, imaginava um futuro onde a tecnologia estaria tão inserida no cotidiano das pessoas que estas não se dariam conta do aparato utilizado. A computação, então, estaria embarcada nos mais diversos dispositivos. Esses conceitos passaram a ser estudados em diversas universidades, em cursos relacionados à Ciência da Computação, que procuravam desenvolver o estudo de tecnologias e sistemas ubíquos. Em geral, esses cursos dispõem de aulas práticas ou mesmo de pequenos projetos de desenvolvimento de aplicações ubíquas. No entanto, umas das maiores dificuldades no ensino de Computação Ubíqua é a prática do conhecimento teórico adquirido, visto que o ferramental existente ou exige fortes conhecimentos de programação ou não foi projetado para fins pedagógicos. Em virtude disso, este trabalho se propõe a conceber, desenvolver e avaliar uma ferramenta que possa auxiliar as aulas práticas de professores que ministram disciplinas com conceitos de Computação Ubíqua. Por meio do *Design* Centrado no Usuário (DCU), os componentes do sistema são testados com usuários reais

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

XXXXXXXXXXXXX – As informações serão preenchidas no processo de edição dos Anais.

a fim de validar a ferramenta proposta. Os resultados das primeiras avaliações com usuários são analisados e apresentados neste artigo.

Categories and Subject Descriptors

Human-centered computing~Empirical studies in ubiquitous and mobile computing.

General Terms

Design, Experimentation, Human Factors, Theory, Verification.

Keywords

Ubiquitous Computing, Learning Environment, Web, Mobile Devices.

Palavras-chave

Computação Ubíqua, Ambiente de Aprendizagem, Web, Dispositivos Móveis.

1. INTRODUÇÃO

O progresso tecnológico contribuiu para o barateamento e a disseminação de diversos dispositivos computacionais (e.g. *laptops*, *smartphones*, sensores) e também de uma variedade de tecnologias de comunicação. Esses avanços proporcionaram a concepção e a forte adoção de serviços computacionais em muitos domínios, tais como agricultura, transporte, medicina e educação.

Esse progresso tecnológico aproxima o ser humano, de uma certa forma, da visão de computação preconizada por Mark Weiser. Em suas pesquisas, que tiveram início em 1988, Weiser imaginava o futuro no qual a tecnologia estaria tão inserida no cotidiano das pessoas que ela seria indistinguível no ambiente, misturando-se com objetos do dia a dia [1]. Em *“The Computer for the 21st Century”*, Weiser introduziu o termo Computação Ubíqua. Neste novo paradigma, os sistemas computacionais estariam “embutidos” em elementos da vida diária, sendo capazes de trocar dados entre si com a menor intervenção possível de usuários [1].

Computação Ubíqua, segundo John Krumm, também se caracteriza por ser um campo de conhecimento multidisciplinar, que envolve a integração de conjuntos de variadas tecnologias usadas para a solução de um desafio específico [2]. Ela é um paradigma que envolve várias áreas de Ciências da Computação, compreendendo sistemas distribuídos [3], voláteis [4], heterogêneos [5] e centrados no usuário [1]. Esse usuário interage com interfaces de forma natural e sem se dar conta da quantidade de computadores, sensores e atuadores espalhados pelo ambiente, da mesma forma que não imagina a interação/integração entre esses componentes, estando a tecnologia completamente inserida no seu dia a dia de forma calma e invisível.

Devido ao potencial tecnológico e de pesquisa da Computação Ubíqua, diversos conceitos desta área são estudados nas universidades pelo mundo, tais como: captura de experiências e

intenções, adaptabilidade, descentralização, descoberta de serviços, heterogeneidade e onipresença [6]. Isso tanto nas áreas de Ciência da Computação, como em áreas afins. Em geral, o objetivo é que esses conceitos sejam assimilados pelos estudantes e que novos sistemas possam ser concebidos e desenvolvidos para auxiliar as pessoas em suas tarefas cotidianas, de forma que estes sistemas se dissolvam na percepção de quem os usa.

No entanto, um dos maiores desafios, para a prática de conceitos de Computação Ubíqua em sala de aula, é a utilização de ferramentas de apoio ao ensino. Isso porque estas são muito escassas, fugazes e, às vezes, de difícil acesso. Os altos custos do *hardware* utilizado, por exemplo, como o LEGO Mindstorms [7], caracteriza uma das dificuldades encontradas.

Embora existam na literatura diversos relatos de experiências no ensino de conceitos de Computação Ubíqua, tais como [8], [9], [7], [10], em geral, as ferramentas ou abordagens utilizadas incluem artefatos ou *softwares* que não foram concebidos especificamente para o ensino de Computação Ubíqua. Como exemplo, podem ser citados o Node-RED [11], voltado para Internet das Coisas, e o App Inventor [12], voltado para o ensino de lógica de programação, ou mesmo ambientes de programação tradicionais de aplicações móveis (e.g. Android Studio).

Segundo [10], em algumas universidades, o interesse em determinadas plataformas e em linguagens de programação, assim como o conhecimento prévio dos estudantes, podem levar a diferentes escolhas para a utilização de artefatos que apoiem as aulas práticas. Por exemplo, o professor pode optar pelo uso do Arduino¹ em cursos nos quais os estudantes estejam mais focados em processamento de dados ou em Interação Humano-Computador (IHC). *Kinect* e outros dispositivos de jogos para aqueles que trabalham com jogos dentro do currículo acadêmico. Ou ainda *smartphones* e seus sensores para quando existir uma plataforma base condizente com a experiência de programação do aluno.

É nesse contexto que esta pesquisa busca melhorar o ensino-aprendizagem de conceitos de Computação Ubíqua por meio da concepção, desenvolvimento e avaliação de uma ferramenta que fará uso de *smartphones* e seus sensores. A escolha destes artefatos ocorre em virtude deles serem mais acessíveis financeiramente e mais utilizados pela maioria dos estudantes. Essa ferramenta consistirá de duas partes: uma aplicação Android, que terá acesso aos sensores do dispositivo e simulará conceitos de Computação Ubíqua; e uma aplicação Web, na qual as configurações da simulação serão definidas tanto pelo professor quanto pelo aluno.

O restante deste documento está organizado em oito sessões que estão descritas a seguir. A Seção 2 traz a fundamentação teórica e os trabalhos relacionados. A Seção 3 aborda a ferramenta que está sendo proposta para este trabalho. A Seção 4 descreve a metodologia seguida. A Seção 5 diz respeito à elicitação de requisitos com professores e alunos. A Seção 6 ilustra como foi concebido e desenvolvido o primeiro protótipo da ferramenta proposta. A Seção 7 explana como foi o teste desse primeiro protótipo. A Seção 8 apresenta e analisa os resultados obtidos a partir da avaliação do protótipo. A Seção 9 trata das considerações finais.

2. FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

No final da década de 80, Mark Weiser, que era cientista da Xerox PARC na época, cunhou o termo Computação Ubíqua, que definia um novo paradigma da computação. Tinha-se, então, o vislumbre de um futuro, no qual as tecnologias de computação se tornariam totalmente integradas aos objetos, podendo ser utilizadas para auxiliar nas atividades cotidianas, tanto no trabalho como na diversão [2]. Weiser considerava que as tecnologias mais profundas são aquelas que desaparecem. Elas se misturam com os objetos do dia a dia até que se tornam indistinguíveis no ambiente em que estão embarcadas [1].

A realização da Computação Ubíqua tornou-se possível graças aos recursos de *hardware* (e.g. *smartphones*, *tablets*, sensores e outros equipamentos com alto poder de comunicação e processamento) e de *software* disponíveis [13]. Conforme a Figura 1, a Computação Ubíqua é fruto da evolução de alguns outros paradigmas que tiveram início com a computação de *mainframe*. Nesta primeira fase, os computadores de grande porte ocupavam o cenário da época com grande poder de processamento de informação. Posteriormente, veio a fase da computação pessoal, segunda fase, conhecida pelos computadores de pequeno porte e de baixo custo, destinados exclusivamente ao uso pessoal ou a um pequeno grupo de indivíduos. Em seguida, tem-se a terceira fase ou a era da computação distribuída. Esta se caracterizou pela dispersão de componentes de *hardware* e de *software* que se comunicam e executam como um único sistema. Então, na quarta fase ou fase atual, há a concretização da Computação Ubíqua. [14]

A Computação Ubíqua se difere da Computação Pervasiva no que se refere a manutenção da conexão e configuração de serviços necessários mesmo durante a mobilidade do usuário. O conceito de pervasividade diz respeito a embarcação de computadores no ambiente de forma invisível ao usuário e sem garantia de muita mobilidade [15]. Apesar dessas diferenças, a definição dos dois conceitos se aproxima bastante, o que leva este trabalho a considerar Computação Ubíqua e Pervasiva como sinônimas.

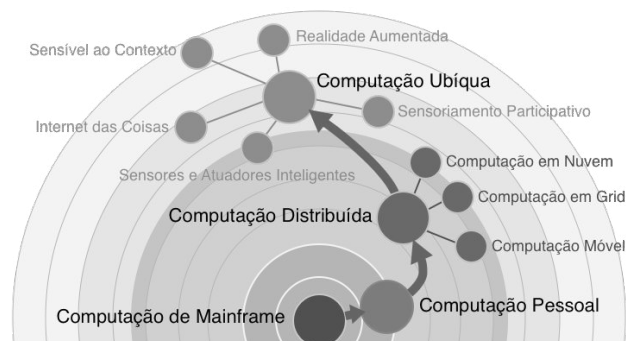


Figura 1. Evolução da Computação Ubíqua Adaptado de [14]

2.1 Principais Conceitos e Características de Computação Ubíqua

Com base em [6] e em [13], após observados alguns projetos de sistemas ubíquos, foram levantadas características fundamentais

¹ <https://www.arduino.cc/>

para a descrição dos conceitos proeminentes da Computação Ubíqua. Essas características são, segundo os autores [6] [13]:

1. **Adaptabilidade:** capacidade do sistema se adaptar às mudanças do ambiente e dos dispositivos para realizar as tarefas requeridas pelo usuário de forma otimizada;
2. **Descentralização:** característica que permite a distribuição das responsabilidades entre diversos dispositivos espalhados no ambiente. Cada dispositivo tem sua função específica;
3. **Descoberta automática de serviços:** os serviços disponíveis no ambiente são descobertos de forma automática pelos dispositivos dos usuários, que podem fazer escolhas e configurações manuais nesses serviços;
4. **Escalabilidade:** capacidade do sistema permanecer funcionando de forma efetiva mesmo diante do aumento do número de recursos ou usuários;
5. **Heterogeneidade:** capacidade do sistema suportar uma grande variedade de dispositivos e serviços sem comprometer seu funcionamento;
6. **Interoperabilidade:** capacidade dos sistemas se comunicarem, sem a intervenção de usuários, para a troca de informações ou de dados;
7. **Invisibilidade:** capacidade dos sistemas se tornarem invisíveis para os usuários, seja pela sua incorporação em objetos do dia-a-dia ou pela realização de suas funções sem chamar a atenção de quem os usa;
8. **Onipresença:** capacidade dos serviços e sistemas se moverem, inclusive com a troca de dispositivos, dando a sensação ao usuário de estar usando o mesmo serviço ou sistema;
9. **Sensibilidade ao contexto:** característica que permite um sistema inferir as necessidades do usuário sem que seja necessário determiná-las explicitamente, conforme o contexto em que o usuário está inserido;
10. **Tolerância a falhas:** capacidade do sistema se recuperar após falhas sem deixar que o usuário perceba.

A Computação Ubíqua concretiza-se através de sistemas computacionais que possuem componentes de *hardware* e de *software* integrados para a realização de tarefas, por meio da execução coordenada de atividades, a qual inclui troca de informações, invocações de serviços e compartilhamento de recursos [6]. Pode-se, então, experimentar as idealizações feitas por Weiser e que não eram possíveis em seu tempo, tais como relógios inteligentes que medem os batimentos cardíacos, smart TVs, que podem ser conectadas à Internet e controladas por smartphones, sistemas de recomendação e espaços multimídia interativos.

2.2 A Computação Ubíqua em Sala de Aula

Conceitos de Computação Ubíqua são vistos em diferentes disciplinas nas universidades. Pode-se citar Redes Móveis Sem Fio, Sistemas Distribuídos, Interação Humano-Computador, Internet

das Coisas ou mesmo aquelas dedicadas exclusivamente ao ensino de Computação Ubíqua.

Em [10], por exemplo, descrevem um cenário de organização de disciplinas que possuem conceitos e práticas de Computação Ubíqua na Universidade de Sussex, no Reino Unido. Essas aulas estão organizadas em três partes: exposição teórica, seminários e práticas em laboratório [10].

Na exposição teórica, são apresentados os conceitos fundamentais de Computação Ubíqua e trabalhos que abrangem, de forma mais generalizada, o estado da arte relacionado ao tema. No momento da apresentação de seminários, o foco é analisar criticamente as teorias e as soluções existentes para resolução de problemas relacionados aos conceitos ubíquos.

Nas práticas em laboratório, o objetivo é programar variados dispositivos, para justificar o entendimento do que foi visto na teoria. O foco é a concepção, a execução e a análise de experimentos, com a posterior apresentação dos resultados obtidos.

Em alguns cursos pesquisados, as disciplinas relacionadas à Computação Ubíqua possuem estruturas similares a apresentada por Chalmers [10], com as práticas sendo realizadas na maior parte do tempo em Android e, eventualmente, em PhoneGap².

A seguir, alguns trabalhos relacionados diretamente ao tema desta pesquisa são listados. Optou-se por aqueles que trazem modelos de curso e sistemas para o desenvolvimento das aulas práticas.

2.2.1 Introdução à Computação Ubíqua Usando SenseBoard

Em [8], tem-se a descrição de um módulo, *My Digital Life* (TU100), criado para auxiliar estudantes novatos da graduação, da educação a distância, no desenvolvimento de programas ubíquos. Eles experimentam, de maneira lúdica, a prática de conceitos de Computação Ubíqua por meio de um *hardware* chamado de SenseBoard, que pode ser programado com a linguagem Sense, uma linguagem em blocos semelhante ao Scratch³.

O dispositivo SenseBoard, citado em [8], é uma placa, com *input*, *output* e alguns sensores, baseado no microcontrolador Arduino⁴, conforme mostra a Figura 2.

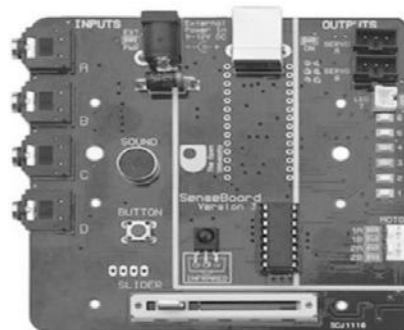


Figura 2. SenseBoard
Fonte: [8]

O SenseBoard surgiu devido à necessidade de uma ferramenta mais simples do que Arduino ou Phidgets, por exemplo, para alinhar a compreensão dos alunos em disciplinas de computação com

² <http://phonegap.com/>

³ <https://scratch.mit.edu/projects/editor/>

⁴ <https://www.arduino.cc/en/Guide/Introduction>

experiências de Computação Ubíqua no mundo real. A ideia era que não fossem exigidos muitos conhecimentos prévios, como em eletrônica, em circuitos ou em linguagem de programação. Isso possibilitou o uso dessa ferramenta em aulas práticas à distância por alunos novatos da *Open University*.

Como o público-alvo de [8] é de alunos novatos, não se poderia exigir que estes já trouxessem uma bagagem de conhecimento em linguagens de programação. Por isso, foi adotada uma extensão da linguagem Scratch, o Sense, que se baseia na estrutura de ligação de blocos.

É possível ver que [8] trabalha o ensino de conceitos de Computação Ubíqua por meio da experimentação prática desses conceitos, sem exigências de conhecimentos prévios em programação. No entanto, o ponto negativo é que, aparentemente, o trabalho foi descontinuado, pois o *hardware* SenseBoard não está mais disponível para a venda e a linguagem Sense, apesar de poder ser baixada, apresenta-se bastante instável, parando de funcionar com muita frequência.

2.2.2 Programação Interativa Pervasiva - PIP

Os autores em [9] apresentam uma ferramenta (Programação Interativa Pervasiva (PIP)) voltada para o ensino de habilidades elementares de programação em conjunto com conceitos de *Internet of Things* (IoT). O PIP é um cenário que envolve a programação de um conjunto de “coisas” para fornecer alguma funcionalidade coordenada (e.g. um reprodutor de música para de tocar no momento em que um telefone começa a chamar). Assim, os estudantes aprendem conceitos básicos de lógica de programação (e.g. *if/else*) por meio da escrita de regras.

Como exemplo de uma prática com o PIP, conforme ilustra a Figura 3, tem-se que três componentes devem ser controlados: um telefone, uma lâmpada e um *media player*. Para isso, as regras devem ser escritas de tal forma que, quando o telefone chamar, as luzes aumentarão o brilho e o *media player* parará de tocar.

O PIP poderia ser usado para o ensino de alguns conceitos ubíquos a estudantes sem muita bagagem de programação. No entanto, ficaria mais limitado à prática da programação e da comunicação entre sensores e objetos. A ferramenta, contudo, não foi encontrada para a realização de possíveis testes.

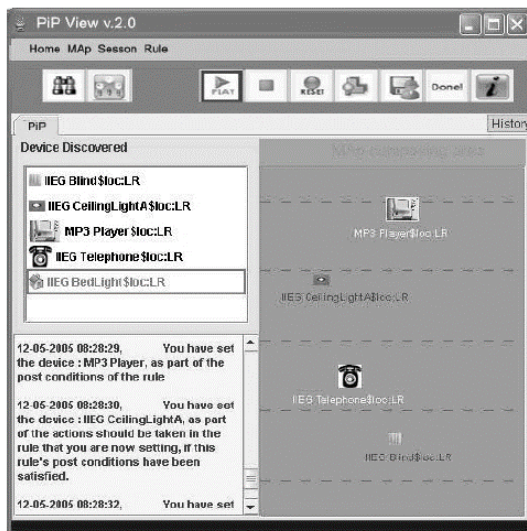


Figura 3. PIP
Fonte: [9]

2.2.3 Ensino de Computação Pervasiva para Alunos de Ciência da Computação

Em [7], existe a preocupação de ensinar Computação Pervasiva a estudantes do primeiro ano do curso de Ciência da Computação. O termo Computação Pervasiva é sinônimo, atualmente, de Computação Ubíqua. A ideia desse trabalho é que esse ensino seja por meio do desenvolvimento de projetos pervasivos, que possam inferir o contexto dos alunos através do reconhecimento de padrões, por exemplo.

Além de projetar e desenvolver os sistemas pervasivos, os alunos são levados a demonstrar as funcionalidades de seus projetos e como os fizeram. Para fornecer suporte a esses projetos práticos, como ilustra a Figura 4, existem os laboratórios equipados com LEGO Mindstorms NXT 2.0 para a parte física a ser controlada. Além disso, eles dispõem do ambiente de trabalho matemático MATLAB, sendo usado como meio de programação para controlar o LEGO NXT.

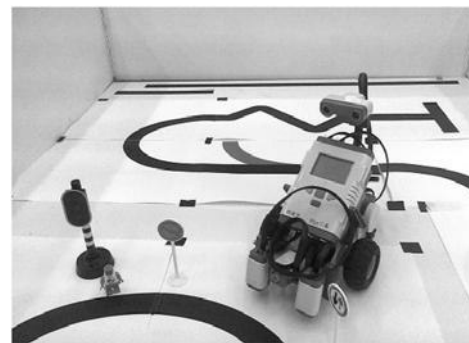


Figura 4. Exemplo de aula em laboratório
Fonte: [7]

Apesar de ser um *hardware* robusto, fácil de programar, móvel e escalável, os pontos negativos do LEGO são que ele possui o custo alto e limitação no que diz respeito a seus sensores e atuadores. Além disso, existe uma forte limitação do tipo de sistema pervasivo que pode ser construído.

2.2.4 Um Curso de Computação Ubíqua

Em [10], é apresentada a estrutura básica de um curso de Computação Ubíqua e, dentro dessa estrutura, estão as aulas práticas em laboratório, que duram em média duas horas. A teoria e a prática são executadas em paralelo e, mesmo que ligadas, ainda funcionam de maneira independente uma da outra.

Dispositivos chamados Phidgets são usados nas aulas práticas, conforme ilustra a Figura 5. Esses possuem APIs fornecidas em várias linguagens de programação. Por conta disso, os alunos, com habilidades em outras linguagens de programação, podem realizar a parte prática do trabalho sem maiores restrições, mesmo que a linguagem padrão escolhida pelo curso seja Java, por exemplo. Isso os permite se concentrar melhor no tema proposto pelo projeto, sem se preocupar com o aprendizado de alguma nova linguagem.

Apesar de existirem equipamentos semelhantes que pudessem substituir os Phidgets, como Motes e Arduino, o autor do trabalho não optou pelo uso destes. Essa escolha decorre do fato de que o uso de Motes ou Arduinos causaria uma mudança no foco da disciplina, que passaria a requerer dos estudantes maior conhecimento de programação em C e programação de *hardwares* embarcados.

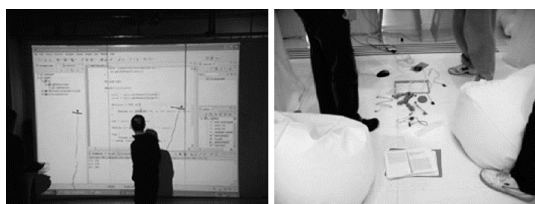


Figura 5. Exemplo de aula prática utilizando Phidgets
Fonte: [10]

O ponto negativo desse trabalho é o custo médio para aquisição dos dispositivos utilizados e a ausência de um ambiente de apoio às aulas práticas para auxiliar na visualização dos conceitos apresentados.

2.3 Análise Crítica dos Trabalhos Relacionados

Os trabalhos relacionados foram resumidos na Tabela 1, na qual cada trabalho relacionado analisado anteriormente representa uma coluna. É possível visualizar os projetos já desenvolvidos e suas características, além das semelhanças e das diferenças entre eles.

Alguns dos projetos apresentados trabalham com *hardware* e *software* feitos, inicialmente, para outras finalidades. Isso significa que esses artefatos podem não abordar os conceitos de Computação Ubíqua diretamente, o que pode exigir mais empenho e criatividade dos professores para a elaboração de atividades práticas. Observa-se então a utilização de kits SenseBoard, ferramentas Web, LEGO Mindstorms NXT, MATLAB e Phidgets. Não foram encontrados, até o momento, trabalhos acadêmicos que utilizem os *smartphones* como artefato para as aulas práticas de Computação Ubíqua, apesar desse dispositivo ser bem popular entre os estudantes.

Com os avanços da pesquisa nas áreas de Computação Móvel e Redes Sem Fio, houve grande popularização no uso de diversos dispositivos nos últimos anos, tais como *tablets*, *smartphones* e *smartwatches*, por exemplo. Isso gerou maior demanda por aplicações voltadas para esses dispositivos, assim como também se passou a exigir maior poder de processamento desses artefatos móveis [16]. Levando em consideração [17], é possível afirmar que o sucesso e a popularidade de aplicações e serviços móveis resultaram em um aumento ainda mais significativo no número de dispositivos móveis no mundo inteiro. Esses dispositivos já se tornaram parte essencial na vida das pessoas, que os usam para os mais diversos fins e em diferentes contextos. Com base no exposto, torna-se mais viável a utilização de *smartphones* para o apoio às aulas práticas de Computação Ubíqua.

Isso devido à maioria dos estudantes possuir tal objeto, além do professor também poder dispor, com menos dificuldades, de alguns aparelhos para a realização dessas práticas.

3. A FERRAMENTA PROPOSTA

O objetivo da ferramenta proposta é contribuir para a seguinte questão de pesquisa:

QP1: Quais características devem existir em uma ferramenta de apoio ao ensino de Computação Ubíqua de forma a melhorar o entendimento dos conceitos teóricos por parte dos estudantes?

Este trabalho visa conceber, desenvolver e avaliar um ambiente Web, sincronizado com uma aplicação móvel Android, conforme ilustra a Figura 6. Esse ambiente pode auxiliar professores e alunos na prática de conceitos teóricos de Computação Ubíqua. Essa ferramenta, em um primeiro instante, foca em algumas características de Computação Ubíqua, tais como adaptabilidade, descoberta automática de serviços, heterogeneidade, interoperabilidade e sensibilidade ao contexto. Para cada um destes conceitos, são disponibilizadas seções com informações teóricas, exemplos de implementações, simulações das camadas de um sistema ubíquo e práticas de programação relacionadas a cada conceito estudado.

A ideia é que para cada conceito estudado (e.g. heterogeneidade e sensibilidade ao contexto) sejam criados Objetos de Aprendizagem (OA) que exemplifiquem o funcionamento desses conceitos. Os alunos poderão visualizar melhor o fluxo da interação entre as camadas que compõem os sistemas ubíquos, enxergando como funcionam a comunicação e o compartilhamento de informações entre sistemas, ambientes e pessoas. Espera-se facilitar o entendimento dos conceitos de Computação Ubíqua com a ferramenta proposta, visando aumentar a criação de aplicações ubíquas e facilitar a utilização de mais dispositivos, que se conectarão a outros dispositivos, pessoas, coisas e lugares. Será considerado, neste trabalho, a noção de Objeto de Aprendizagem como:

“(…) recursos digitais dinâmicos, interativos e reutilizáveis em diferentes ambientes de aprendizagem elaborados a partir de uma base tecnológica. Desenvolvidos com fins educacionais, eles cobrem diversas modalidades de ensino: presencial, híbrida ou a distância; diversos campos de atuação: educação formal, corporativa ou informal; e, devem reunir várias características, como durabilidade, facilidade para atualização, flexibilidade, interoperabilidade, modularidade, portabilidade, entre outras” [18]

Tabela 1. Trabalhos Relacionados.

	SenseBoard [8]	PIP [9]	Pervasive Computing [7]	Pervasive Computing Classroom [10]
Tipo	Desktop e hardware	Desktop	Desktop e hardware	Hardware
Público-Alvo	Graduação - EAD	Sem restrições	Graduação	Pós-graduação
Nível do Público	Sem exigências	Sem exigências	Sem exigências	Conhecimentos de programação
Objetivo	Explorar Computação Ubíqua por meio de experimentos práticos	Ensinar conceitos de programação	Ensinar Computação Pervasiva através de projetos	Ensinar Computação Pervasiva por meio da prática em laboratório
Recursos Exigidos	Kit SenseBoard e Linguagem Sense	Ferramenta PIP instalada	LEGO Mindstorms NXT e MATLAB	Phidgets
Custo Financeiro	Médio/alto	Baixo	Alto	Médio/alto
Fácil Disponibilidade	Kit não disponível para compra ⁵ ; Linguagem disponível ⁶	Material não disponibilizado e não encontrado	LEGO NXT 2.0 ⁷ não disponível para compra; MATLAB ⁸ disponível	Sim para Phidgets ⁹

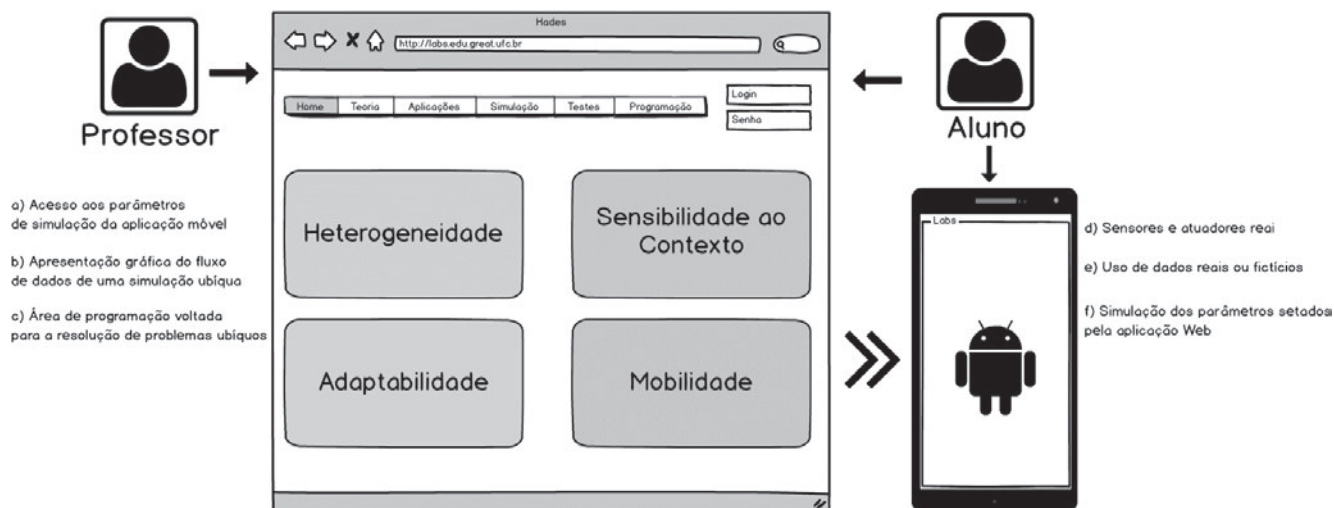


Figura 6. Características das aplicações

Conforme a Figura 6, a parte Web da ferramenta possui parâmetros de configuração da aplicação móvel, apresenta um fluxograma dos dados de uma aplicação ubíqua e possui área voltada à programação. Alunos e professores possuem acesso diferenciado à essa parte Web, pois o professor faz as configurações para serem visualizadas nos celulares de todos os estudantes, enquanto os alunos visualizam suas próprias configurações em seus próprios aparelhos. A parte móvel traz acesso a sensores e atuadores do próprio dispositivo utilizado, usa dados reais e simula o comportamento das configurações feitas no ambiente Web.

4. METODOLOGIA

A metodologia utilizada neste trabalho é a do *Design Centrado no Usuário (DCU)*, na qual o desenvolvedor não só imagina como será a interação do usuário final com o sistema que será criado, mas também testa essa mesma aplicação com usuários reais para dar validade ao trabalho. Segundo [19], o DCU é a prática de criar produtos de forma que os usuários sejam capazes de utilizá-los com o mínimo de *stress* e o máximo de eficiência.

Essa metodologia tem se mostrado bastante eficaz para a concepção de sistemas, pois destaca a importância do aperfeiçoamento e refinamento do projeto por meio dos testes de prototipagem [20].

O DCU foi escolhido para validar melhor a ferramenta que será criada, a fim de evitar excesso de informações e funcionalidades inúteis, deixando essa ferramenta de acordo com as necessidades reais dos usuários finais. As etapas do DCU estão divididas em quatro partes, como mostra a Figura 7.

A primeira parte é identificar os requisitos dos usuários e do cliente. Isso foi feito por meio de entrevistas, guiadas por um questionário

preestabelecido, a professores que ensinam ou ensinaram, em algum momento, conceitos de Computação Ubíqua em suas disciplinas do ensino superior (graduação e pós-graduação). Além disso, também foi aplicado um questionário aos estudantes que fizeram alguma disciplina com conceitos de Computação Ubíqua.

Duas modalidades diferentes de levantamento de requisitos (entrevista e questionário) foram escolhidas para serem aplicadas com professores e com alunos. Isso pelo fato de ser necessário identificar e perceber melhor do cliente (os professores) a estrutura e organização das aulas, assim também como eram feitas as práticas em sala de aula e quais as maiores dificuldades. Por isso, optou-se pela realização de entrevistas, devido ao nível qualitativo dos dados, com os professores, conforme o guia de [14].

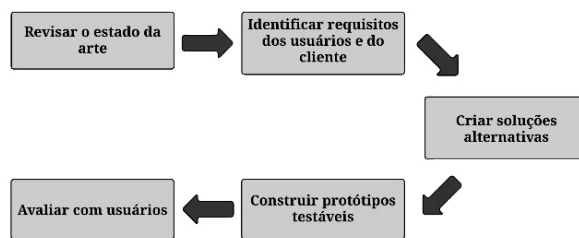


Figura 7. Metodologia de pesquisa inspirada no DCU
Fonte: [21]

Para os estudantes, que representam parte dos usuários do projeto, foi escolhida a aplicação de questionário com perguntas de maioria múltipla escolha. O intuito era facilitar o preenchimento e gerar maior número de dados.

⁵www.ouw.co.uk/dp/178007395X

⁶www.open.edu/openlearnworks/mod/oucontent/view.php?id=22780%3F

⁷shop.lego.com/en-US/LEGO-MINDSTORMS-NXT-2-0-8547

⁸www.mathworks.com/downloads

⁹www.phidgets.com/

A segunda etapa do DCU é o momento de criar soluções alternativas, que foram desenvolvidas depois do estudo e da avaliação de vários trabalhos. A partir de algumas ideias postas no papel, foi construído o primeiro protótipo testável, consolidando a terceira fase do DCU. Em seguida, a avaliação do protótipo com os usuários finais foi feita, para que o trabalho pudesse ser validado e os resultados apresentados.

Das etapas do Design Centrado no Usuário, tem-se, no início, a fase de revisão do estado da arte, na qual pesquisas e estudos de trabalhos semelhantes à proposta e relacionados ao tema de Computação Ubíqua foram analisados.

5. ELICITAÇÃO DE REQUISITOS

5.1 Professores

Algumas entrevistas com professores foram feitas a fim de se conhecer quais conceitos, na visão deles, eram considerados mais importantes. Também procurou-se identificar a experiência do professor, a organização do curso, os materiais usados em aulas práticas, o número de práticas, as dificuldades relacionadas ao ensino de conceitos de Computação Ubíqua e como eles imaginam um ambiente para a prática de Computação Ubíqua. Assim, foram entrevistados três professores do Brasil e uma professora brasileira que leciona em uma universidade da França.

As entrevistas foram feitas de forma presencial ou pelo aplicativo *Skype* e guiadas por um questionário¹⁰. Elas levaram, em média, trinta minutos cada e foram gravadas com ciência e permissão dos participantes. Os entrevistados são identificados neste trabalho como Professor 1, Professor 2, Professor 3 e Professor 4 com o intuito de garantir o sigilo da identidade dos participantes.

Dentre as questões apresentadas no questionário-guia, apenas as de maior relevância para a análise dos dados estão discriminadas a seguir:

1. Você utiliza algum ambiente/kit para ensinar Computação Ubíqua? Se sim, qual?
 - **Professor 1** = Android Studio;
 - **Professor 2** = Não utiliza um ambiente específico, mas vários;
 - **Professor 3** = Android Studio;
 - **Professor 4** = Não utiliza.
2. Quais as dificuldades que você encontra no ensino dos conceitos de Computação Ubíqua?
 - **Professor 1** = Muito conteúdo para pouco tempo e aplicação da teoria na prática;
 - **Professor 2** = Aplicar teoria na prática;
 - **Professor 3** = Complexidade dos conceitos de Computação Ubíqua, dificuldade de ver como os conceitos teóricos são aplicados na prática;
 - **Professor 4** = Muito conteúdo para pouco tempo.
3. Na sua opinião, o que esse ambiente interativo deveria ter?

- **Professor 1** = Comportamento de informações contextuais e teoria;
- **Professor 2** = Simulação do comportamento de dispositivos e atuação de sensores;
- **Professor 3** = Conceitos de sensibilidade ao contexto, como modelar informações contextuais importantes para a aplicação, adaptação da aplicação conforme as informações de contexto;
- **Professor 4** = Vídeos, tutoriais que expliquem o funcionamento de ferramentas já existentes (e.g. *arduino* ou *raspberry*) e OA's que incrementem a aprendizagem.

4. Qual o conceito de Computação Ubíqua você considera como o mais importante para o ambiente interativo?
 - Todos os professores elegeram a Sensibilidade ao Contexto.

Diante das respostas, pode-se perceber que a maioria dos professores não possui um ambiente específico para o ensino de Computação Ubíqua, utilizando, algumas vezes, o *Android Studio*. Isso contribui para que o ambiente proposto adote, como plataforma base, o próprio *Android*, por ser gratuito e usado por boa parte dos professores.

As três últimas perguntas são extremamente relevantes para esta pesquisa. Elas apontam as maiores dificuldades que os professores têm em sala de aula, o que eles imaginam que o ambiente proposto por este trabalho tenha e o conceito considerado por eles como o de maior relevância para o ensino de Computação Ubíqua.

5.2 Estudantes

Foi também aplicado um questionário¹¹ com alguns alunos. Ele era composto por onze questões. Esse questionário foi aplicado de forma *online*, via grupos de disciplinas no *Facebook* que tinham relação direta com Computação Ubíqua ou que eram exclusivamente sobre o assunto.

O objetivo do questionário foi saber dos estudantes se eles possuíam experiência de programação no momento da realização da disciplina, se tinham aulas práticas de Computação Ubíqua, se usavam alguma ferramenta para as aulas práticas, quais os conceitos que eles mais tinham dificuldade de entender e o que eles esperavam do ambiente proposto.

Depois da disponibilização dos questionários em grupos de disciplinas no *Facebook*, obteve-se o total de trinta respostas. Os alunos, em geral, eram de cursos ligados à tecnologia de algumas universidades brasileiras. Uma análise mais detalhada das principais respostas pode ser vista a seguir.

Os resultados dos questionários indicam que a maioria dos estudantes tiveram aulas práticas dedicadas à Computação Ubíqua. Quando perguntado o que eles utilizavam nessas práticas, a maioria respondeu que usava o *Android Studio* como ambiente. Tem-se então um ambiente/plataforma de maior difusão devido a, provavelmente, sua gratuidade e a sua maior adesão, por usuários, no mercado de dispositivos móveis.

Uma outra pergunta foi relacionada aos conceitos mais difíceis de se entender na ausência de aulas práticas. A maioria dos estudantes

¹⁰ <https://goo.gl/forms/MHdyKuCtnBk3whx1>

¹¹ <https://goo.gl/forms/f7j2IUeCtOfmpBP73>

respondeu que o conceito de “*middleware*” foi o mais difícil de se entender, seguido por “*adaptação*” e “*sensibilidade ao contexto*”, conforme ilustra a Figura 8.

Quando perguntados sobre o que deveria ter no ambiente proposto por este trabalho, a maioria respondeu que, como mostrado na Figura 9, deveria haver “*simulação do comportamento de sensores e serviços a partir da inserção de parâmetros*”. Na sequência, falaram de “*programação de códigos Android, pré-moldados pelo ambiente, para visualização, em dispositivos móveis, dos conceitos ubíquos*” e “*representação gráfica dos conceitos teóricos*”.

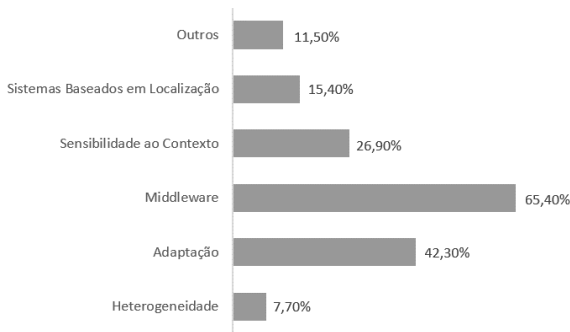


Figura 8. Conceitos difíceis de se entender na ausência de aulas práticas por estudantes

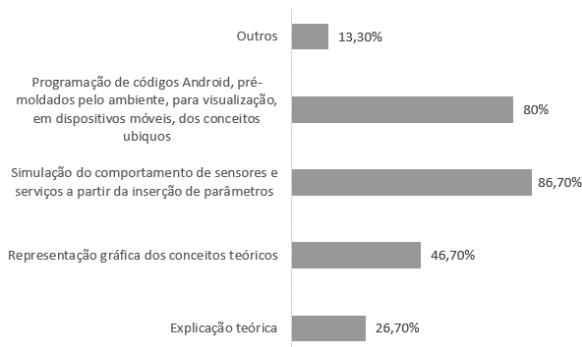


Figura 9. Features do ambiente segundo os estudantes

6. CONCEPÇÃO E DESENVOLVIMENTO DO PRIMEIRO PROTÓTIPO

A partir da elicitação de requisitos, o primeiro protótipo começou a ser delineado. A proposta do ambiente em sua totalidade é que para cada conceito de Computação Ubíqua estudado, haja um Objeto de Aprendizagem (OA) que possa ajudar no seu entendimento. Para a estruturação desse primeiro OA, escolheu-se o conceito considerado como mais importante pelos professores: *sensibilidade ao contexto*. A escolha desse conceito, também permite auxiliar os estudantes na introdução daquele que eles julgam mais difícil de se entender: *middleware*. Isso porque para explicar *sensibilidade ao contexto*, propõe-se mostrar um fluxo de dados em camadas. Nessa estruturação de camadas, algumas delas teriam uma função semelhante ao de *middleware*, mediando e interpretando a informação que é recebida do ambiente e apresentada ao usuário por uma aplicação móvel.

A primeira versão do OA requer configurações para executar alguns comportamentos de um sistema sensível ao contexto. Em outras palavras, escolhas são feitas no ambiente Web a fim de fazer a aplicação móvel capturar e usar informações de contexto do

usuário, adaptando o seu comportamento ao que foi especificado [22].

6.1 Arquitetura do Protótipo

O primeiro protótipo do OA sobre *sensibilidade ao contexto* divide-se em duas partes: Web e móvel. Para a comunicação entre esses dois componentes, foi criado um servidor Web em Ruby. Esse servidor estabelece a comunicação via *WebSocket* dentro de uma mesma rede *wi-fi*. Assim, a porta e o endereço IP da máquina onde está rodando o cliente Web são colocados no cliente móvel. A conexão entre as duas partes da ferramenta pode ser de um cliente Web para vários móveis ou de um cliente Web para um móvel. Isso permite o professor executar uma determinada atividade prática para visualização nos dispositivos de todos os alunos ao mesmo tempo ou cada estudante realiza uma determinada tarefa e verifica o comportamento da ferramenta em seu próprio dispositivo.

6.1.1 Cliente Web

O cliente Web foi desenvolvido utilizando-se HTML e JavaScript. Este cliente traz uma estrutura em camadas, onde sensores podem ser habilitados, desabilitados e configurados de acordo com o contexto do usuário, além da inserção de parâmetros e de tomadas de decisão. Nessa parte da ferramenta, novas regras contextuais também podem ser criadas. Todas as configurações realizadas no cliente Web podem ser visualizadas e testadas no cliente móvel. Da forma que foi pensado e criado, esse componente faz uma analogia com as camadas de um sistema sensível ao contexto do *Context Toolkit* proposto por Dey [23].

Segundo *Context Toolkit* de Dey, conforme a Figura 10, as abstrações realizadas para a construção dessa arquitetura facilitam o entendimento de conceitos para a construção de aplicações sensíveis ao contexto. Conforme essa arquitetura, a camada mais básica é a de sensores, que mostra a disponibilidade de sensores que uma aplicação pode acessar. Mais acima, está a camada de *widgets*, que encapsula uma informação do contexto do usuário, como local ou uma atividade, e gera uma interface para aplicação, escondendo os detalhes de implementação. Depois, tem-se a camada de agregadores, que mescla informações contextuais, como localização e luminosidade, por exemplo, para gerar uma nova informação contextual, como nível de segurança do local. Por último, tem-se os interpretadores, que interpretam informações de contexto de baixo nível, valor de luminosidade em lúmens, transformando-as em informações de nível superior, por exemplo uma barra horizontal que é preenchida ou esvaziada à medida que a luminosidade aumenta ou diminui respectivamente.

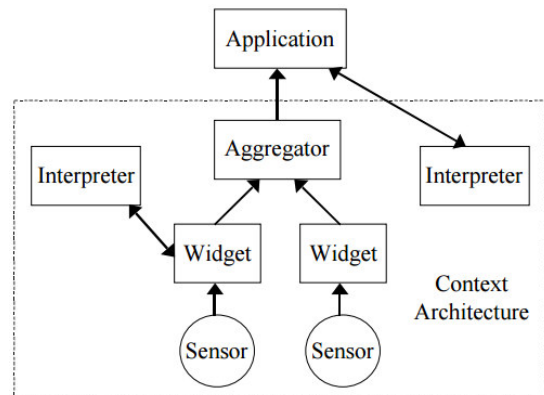


Figura 10. Componentes do *Context Toolkit* proposto por Dey [23]

No ambiente proposto, conforme a Figura 11, o fluxo dos dados pode ser observado por meio da ativação das camadas, que são habilitadas conforme a camada inferior vai sendo ativada pelo aluno ou pelo professor. Tem-se o total de cinco camadas nessa versão Web. A camada de sensores mostra os sensores disponíveis para serem usados, podendo ser habilitados ou desabilitados pelo usuário. Há então GPS, sensor de luminosidade, acelerômetro e sensor de temperatura ambiente rodando em arduino. Nessa primeira versão, o GPS e o sensor de temperatura ainda não foram implementados. No entanto, ao selecionar o sensor de GPS aparecem localizações (coordenadas e nomes de ruas) criadas estaticamente, apenas para serem utilizadas em uma primeira prática em sala de aula. O sensor de temperatura ficou totalmente desabilitado, somente indicando o local do próximo sensor a ser implementado, com o intuito de instigar, nesse primeiro momento, a curiosidade e o debate entre os estudantes.

A camada de *widjets* é responsável por mostrar, na aplicação móvel do usuário, informações adquiridas pelo sensor habilitado. A camada de interpretadores interpreta informação de baixo nível e apresenta uma de nível superior. Os agregadores juntam dados dos sensores para gerar uma nova informação contextual de alto nível. Na tomada de decisão, tanto podem ser utilizados os dados dos sensores como as informações contextuais geradas pelos agregadores.

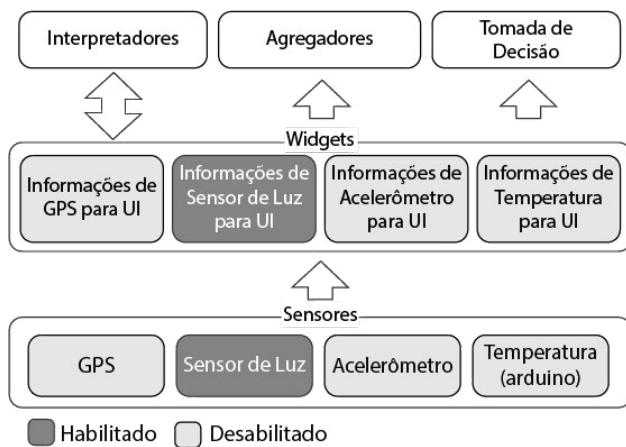


Figura 11. Esquema da Tela do Objeto de Aprendizagem sobre Sensibilidade ao Contexto

A representação da Figura 11 se assemelha a arquitetura do *Context Toolkit* proposto por Dey [23]. Isso porque simplifica, para os alunos, a apresentação do fluxo de dados que uma aplicação sensível ao contexto possui no decorrer das camadas. Na representação proposta por este trabalho, tem-se o acréscimo de mais uma camada, a de tomada de decisões, onde um atuador do celular (e.g. lanterna) pode ser ativado ou desativado conforme a captura de informações configuradas pelo usuário no ambiente Web.

6.1.2 Cliente Móvel

O ambiente móvel foi criado em versão Android. Essa tecnologia foi escolhida por ser gratuita e a mais utilizada pelos professores em suas aulas práticas. Nessa parte da ferramenta, observa-se o comportamento de um sistema sensível ao contexto conforme as configurações feitas no ambiente Web pelo professor ou pelos alunos. O sistema móvel é inicialmente uma tela em branco, que vai sendo preenchida conforme as configurações dos usuários na ferramenta Web. Assim, se o sensor de luminosidade for iniciado dentro da camada de *widjets* no sistema Web, conforme mostra a

Figura 11, valores em lúmens aparecem no sistema móvel, como ilustrado na Figura 12, de acordo com a variação de intensidade da luz do ambiente real captado pelo sensor do celular.

Conforme pode ser visualizado na Figura 12, a aplicação móvel só apresenta as informações, configuradas pelo usuário no ambiente Web, à medida que elas vão sendo adquiridas do seu contexto. Assim, o sistema apresenta, no topo, conforme foi configurado, um indicador para mostrar quais tipos de sensores estão ativos na aplicação e um valor de luminosidade, em lúmens, obtido a partir do sensor do próprio dispositivo. Tem-se também as coordenadas e um mapa de localização. Além disso, há a barra de progresso referente a interpretação de luminosidade, juntamente com um classificador do nível da luz, que varia entre “no light”, “dim”, “low” e “bright”. Mais abaixo, está o interpretador do acelerômetro, que apresenta valores em X, Y e Z à medida que o dispositivo é chacoalhado. Na parte D da Figura 12, tem-se um alerta indicando que, segundo as configurações que o usuário fez no ambiente Web, o local onde ele se encontra é considerado seguro. Da mesma forma que aparece um alerta para indicar que o local do usuário é seguro, também pode aparecer um alerta de local perigoso ou ser ativada a lanterna do dispositivo caso não haja luz. Isso depende das configurações feitas na parte Web da ferramenta.

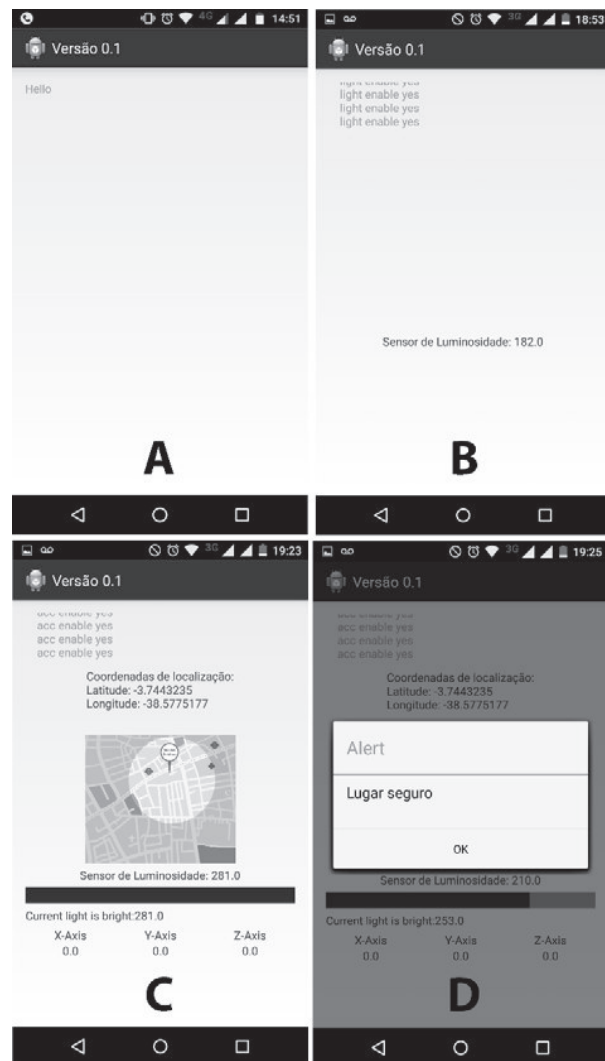


Figura 12. Reflexo da execução no dispositivo móvel das configurações realizadas no ambiente Web

7. TESTE DO PRIMEIRO PROTÓTIPO

7.1 Perfil

O teste do primeiro protótipo, como ilustra a Figura 13, foi realizado numa disciplina de introdução a Computação Móvel e Ubíqua de um curso de graduação de uma universidade pública brasileira. A prática foi feita guiada por um questionário¹² que explicava o que devia ser feito e pedia a apresentação e interpretação dos resultados obtidos através do sistema. A sala foi dividida em dois trios e três duplas, que responderam também, ao final da prática, a um questionário¹³ de avaliação da ferramenta.



Figura 13. Primeiro teste da ferramenta em sala de aula

7.2 Material

Existiam cinco computadores desktop configurados com sistema operacional Linux Ubuntu 15.10, com Python nativo e Ruby 2.3.1, rodando o cliente Web. Para cada uma dessas máquinas, foi gerado um APK com o endereço IP e a porta correspondente a elas. Os estudantes se agruparam nos computadores disponíveis e instalaram, em seus próprios *smartphones*, o aplicativo referente apenas à máquina em que estavam executando a atividade proposta. Os dispositivos utilizados variavam em tamanho, modelo, marca e versão do sistema operacional Android.

7.3 Procedimento

Estabelecida a comunicação dos clientes Web e Android, os alunos foram orientados a seguir um questionário para a realização da atividade prática. Em umas das questões, por exemplo, os estudantes foram solicitados a habilitar o sensor do acelerômetro e descrever o que seu *widget* e interpretador realizavam. Deviam apresentar também *screenshots* do que era mostrado na tela do dispositivo móvel, conforme suas configurações na aplicação Web. Deviam ainda testar os parâmetros de sensibilidade do interpretador do acelerômetro e descrever a funcionalidade deles.

Outros exemplos do passo a passo a ser seguido eram questões como “Combine agora o acelerômetro e a luminosidade para a tomada de decisão. O que aconteceu no aplicativo móvel?”, “Após a realização destas simulações, como você pode diferenciar os *widgets* de interpretadores?” e “Utilize os contextos de alto nível para definir os *triggers* complexos e modifique o seu contexto atual para que a lanterna seja acendida.”

7.4 Instrumento de Avaliação

O questionário era composto de quinze perguntas. Os estudantes tiveram o tempo em aula de uma hora e trinta minutos para conseguir concluir completamente a atividade. Assim, todos os alunos finalizaram com êxito a proposta, além de responderem a avaliação sobre o ambiente.

A avaliação do Objeto de Aprendizagem (OA) se deu logo após seu uso e levou em consideração três aspectos básicos: Pedagógicos,

Requisitos e Interface. Esta abordagem foi proposta por [24], no qual pode ser verificado que os OAs analisados por ele são voltados à parte de Material Didático. Assim, foram feitas algumas adaptações para que se pudesse usar esse modelo aplicado a um *software*.

A abrangência e a simplicidade de avaliação foram as principais vantagens observadas nesse modelo, que é aplicado na forma de questionário para que o público alvo possa responder e, a partir daí, ser feita uma análise sobre os dados. A escala usada no experimento foi a Likert, uma modificação da escala original, que usava simplesmente respostas dicotômicas para as perguntas.

8. APRESENTAÇÃO E ANÁLISE DOS RESULTADOS

Obtiveram-se oito respostas dos alunos que participaram do experimento em sala de aula, alguns responderam o questionário em dupla, outros de forma individual.

Os aspectos pedagógicos dizem respeito a orientação pedagógica do Objeto de Aprendizagem, ou seja, se ele foi construído com diretrizes mais voltadas ao Construtivismo ou Behaviorismo, por exemplo. Além disso, permite saber sobre o uso efetivo do OA do ponto de vista pedagógico na ótica dos alunos. Isso pode ser notado na percepção dos estudantes sobre como esse OA se comporta em relação a eles. Conforme mostra a Figura 14, o OA apresenta diferentes caminhos para solucionar problemas segundo 87,5% dos participantes; corroborando com 62,5% de alunos que não acham que a navegação entre os níveis de dificuldade do OA seja linear e, por fim, 75% concordam que o OA provê informações adaptadas para diferentes estilos de aprendizagem. A Figura 15 ilustra que 62,5% dos participantes, contra 25%, acreditam que o Objeto de Aprendizagem instiga pesquisas em fontes externas a eles. Nesse caso, o OA necessita ainda de mais melhorias para instigar sempre o aluno na busca de fontes de informação externa. Esses fatores juntos, permitem concluir que o OA esteja mais próximo de uma ferramenta que, na percepção dos alunos, seja construtivista. E como 100% dos alunos acham que este OA os instigou a interação em grupo, há um forte indício que ele possa ser usado para ações sócio-interacionistas, do ponto de vista pedagógico. O OA também se mostrou, pelo que foi observado, muito efetivo para o fim a que foi destinado.

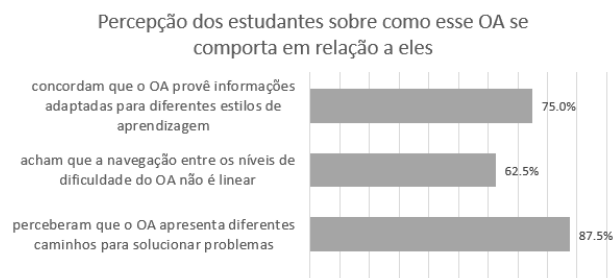


Figura 14. Aspectos Pedagógicos

¹² bit.ly/praticaCompUbiqua

¹³ <https://goo.gl/forms/RYLEvCSacXv91VF52>

O Objeto de Aprendizagem instiga pesquisas em fontes externas aos alunos?

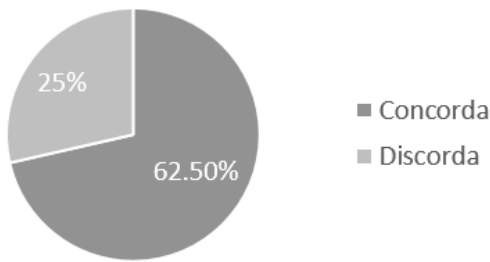


Figura 15. Aspectos Pedagógicos

No aspecto de requisitos, o OA é avaliado quanto a sua robustez, presença de erros, desempenho e suporte a multiplataformas. A Figura 16 mostra que 87,5% dos interagentes observaram algum erro na execução do OA, quer seja na parte Web ou na parte móvel. Em termos de desempenho, 37,5% dos entrevistados acham que o uso intenso do OA vai interferir em sua *performance* e 37,5% acham que não. Ficando 25% sem opinião sobre esta questão. Já quanto a aplicação móvel, 62,5% dos participantes acreditam que o OA possa ser usado em *smartphones* dos mais simples aos mais avançados. Com relação à parte Web, 37,5% dos entrevistados acham que ela pode ser usada em diferentes Sistemas Operacionais, tais como Linux, Mac OS X e Microsoft Windows, ficando 62,5% sem opinar sobre esse quesito. Nesse caso, no aspecto requisitos, o sistema apresenta desempenho mediano para alta escalabilidade, mostrando ainda que satisfaz em multiplataforma e em diversos tipos de dispositivos.

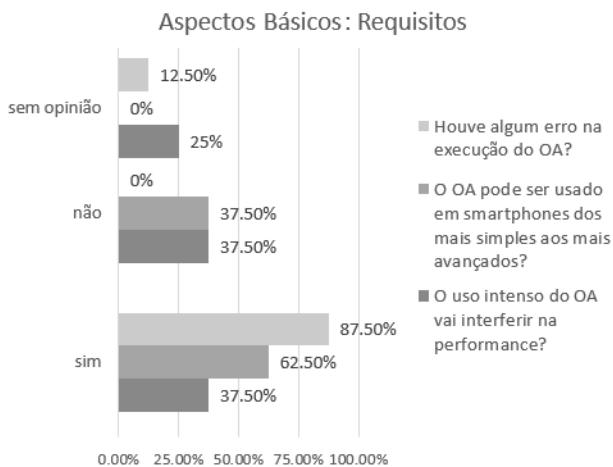


Figura 16. Aspectos de Requisitos

Por fim, foi avaliado o aspecto de interface apresentado pela Figura 17. Quanto ao uso de rótulos e títulos para orientação do interagente, 62,5% dos alunos concordaram que a ferramenta possuía bons recursos e 37,5% não tiveram opinião sobre o assunto. Quanto aos recursos gráficos e elementos gráficos utilizados, 50% dos participantes gostaram e 25% não tiveram opinião sobre o assunto, ficando 25% com opinião de que estes elementos não estariam bons. No geral, portanto, a interface estaria mediana, suprimindo as necessidades básicas de boa parte da turma, mas precisando ser melhorada.

Aspectos Básicos: Interface

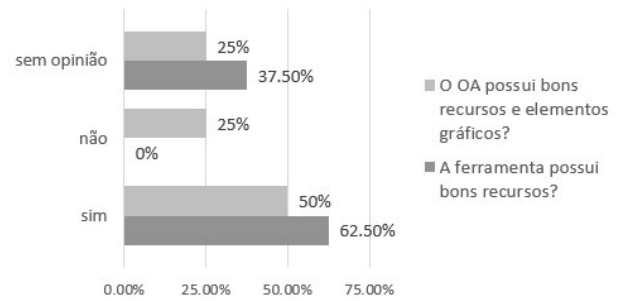


Figura 17. Aspectos de Interface

9. CONSIDERAÇÕES FINAIS

Este trabalho apresenta como foco a concepção, desenvolvimento e avaliação de um ambiente para auxiliar professores e alunos em aulas práticas de Computação Ubíqua. Como já descrito, foi possível verificar, através do levantamento bibliográfico inicial, que existem alguns trabalhos tratando sobre a temática de materiais para o auxílio em aulas práticas de Computação Ubíqua. Porém, eles se utilizam de componentes ou ferramentas com outras finalidades base, o que não supre a maior parte das necessidades das práticas em sala de aula.

Os resultados mostram a importância e a necessidade de uma ferramenta voltada especificamente para o ensino prático dos conceitos de Computação Ubíqua. O ambiente proposto ainda está em fase inicial de desenvolvimento, tendo sido feito somente um Objeto de Aprendizagem para um conceito específico de Computação Ubíqua, que foi o de sensibilidade ao contexto. Nesse OA desenvolvido, foi possível debater e executar diversos exercícios relacionados ao conceito estudado, facilitando a interação entre professor-aluno e aluno-aluno, além do aprendizado compartilhado.

Com a ferramenta proposta, foi possível ilustrar o que se caracteriza como contexto do usuário, como esse contexto é percebido pelas aplicações sensíveis ao contexto e o que estas podem fazer conforme as configurações dos usuários para a variância de contexto. A teoria ensinada em sala de aula passou então a ser exemplificada pela ferramenta proposta, o que gerou mais discussões sobre o tema. Isso também promoveu interações mais frequentes entre professor-aluno e aluno-aluno com a finalidade de sanar dúvidas ou explicar características do conceito estudado.

Pretende-se continuar com as entrevistas e questionários aplicados com professores e alunos, a fim de se ter mais dados que possam contribuir ainda mais com a estruturação do ambiente. Esse primeiro protótipo também será aperfeiçoado para a realização de novas aulas práticas e avaliações em sala de aula. Além disso, mais propostas de conteúdos para a ferramenta serão concebidas e implementadas, assim como novos Objetos de Aprendizagem com outros conceitos de Computação Ubíqua também serão desenvolvidos.

10. AGRADECIMENTOS

A autora principal agradece à FUNCAP pelo apoio financeiro concedido através da cessão de bolsa de mestrado.

11. REFERÊNCIAS

- [1] Weiser, Mark. The computer for the 21st century. *Scientific American*, 265, 3 (#sep# 1991), 66-75.