



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO
MESTRADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO

MARIA JOELMA PEREIRA PEIXOTO

LUCY: UM AMBIENTE PARA AULAS PRÁTICAS DE COMPUTAÇÃO UBÍQUA

FORTALEZA

2017

MARIA JOELMA PEREIRA PEIXOTO

LUCY: UM AMBIENTE PARA AULAS PRÁTICAS DE COMPUTAÇÃO UBÍQUA

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Sistemas de Informação

Orientador: Prof. Dr. Windson Viana de Carvalho

FORTALEZA

2017

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

P4311 Peixoto, Maria Joelma Pereira.

Lucy: um ambiente para aulas práticas de Computação Ubíqua / Maria Joelma Pereira Peixoto. – 2017.
123 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2017.

Orientação: Prof. Dr. Windson Viana de Carvalho.

1. Computação Ubíqua. 2. Ambiente de Aprendizagem. 3. Web. 4. Dispositivos Móveis. 5. Lucy. I. Título.
CDD 005

MARIA JOELMA PEREIRA PEIXOTO

LUCY: UM AMBIENTE PARA AULAS PRÁTICAS DE COMPUTAÇÃO UBÍQUA

Dissertação apresentada ao Curso de Mestrado Acadêmico em Ciência da Computação do Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial à obtenção do título de mestre em Ciência da Computação. Área de Concentração: Sistemas de Informação

Aprovada em: 29 de maio de 2017

BANCA EXAMINADORA

Prof. Dr. Windson Viana de Carvalho (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Fernando Antonio Mota Trinta
Universidade Federal do Ceará (UFC)

Prof.^a Dr.^a Maria de Fatima Costa de Souza
Universidade Federal do Ceará (UFC)

Prof. Dr. Carlos André Guimarães Ferraz
Universidade Federal de Pernambuco (UFPE)

À minha família, por sua capacidade de acreditar e investir em mim. Mãe e pai, seus cuidados e dedicação foram o que me deram, em alguns momentos, a esperança para seguir. Andrei, sua presença significa amor, carinho, paz e certeza de que não estou sozinha nessa caminhada.

AGRADECIMENTOS

Agradeço primeiramente a Deus, pois a fé é a origem da esperança que nos mantém firmes.

À minha família, em especial aos meus pais Maria e José, que me incentivaram a continuar a caminhada e a ser forte a cada dia.

Ao meu esposo Andrei por ser sempre meu ponto de tranquilidade, paciência e amor diante de minhas agitações e nervosismos constantes. Definitivamente, eu te amo demaaaais!

Aos meus irmãos Jaqueline, Jackson, Cléa e Gilson, que me fazem rir e saber que irmãos existem para ensinar que *bullying* começa de casa mesmo, mas também que estarão sempre disponíveis nas necessidades.

Ao meu orientador, professor Dr. Windson Viana, pela paciência e atenção, principalmente, nos meses que se seguiram diante de várias doenças e viroses recorrentes. Obrigada pelos conselhos e incentivos. Se tem uma palavra que vai ficar é: “caaalma”.

Ao meu coorientador, professor Me. Wellington Sarmiento, que não pode ser oficialmente citado como tal, mas foi de grande importância e ajuda no desenvolvimento deste trabalho. Obrigada pela atenção, ajuda e amizade.

Ao professor Marcio Espíndola pelas ideias para o desenvolvimento e aperfeiçoamento deste trabalho. À banca desta dissertação. Ao professor Fernando Trinta, que também auxiliou nos testes finais do ambiente. À professora Fátima Souza. Ao professor Carlos Ferraz. Muito obrigada por contribuírem com o desenvolvimento deste trabalho no decorrer do mestrado.

Aos colegas e amigos do programa de mestrado, em especial ao Paulo Artur, por sua disponibilidade a ajudar sempre que solicitado; ao Felipe Barreto, pela paciência e ajuda quando procurado por dúvidas em Android; e ao Pedro Italo pela ajuda no *design* e no *layout* do ambiente.

E à Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico (Funcap) pelo financiamento desta pesquisa de mestrado via bolsa de estudos.

“Não devo ter medo. O medo mata o pensamento. O medo é a morte antecipada que nos aniquila. Hei de encarar meu medo, deixar que ele me perpassse, que me atravessasse, e quando houver passado, olharei bem dentro de mim para conferir sua trajetória. Por onde passou o medo, não haverá nada. . . resto apenas eu.”

(Frank Herbert)

RESUMO

Mark Weiser cunhou o termo Ubiquitous Computing (UbiComp) descrevendo um futuro no qual objetos de vida cotidiana teriam computadores incorporados fornecendo serviços a qualquer hora e em qualquer lugar. Uma das atividades mais difíceis no ensino do UbiComp é colocar em prática o conhecimento teórico, já que as ferramentas atuais de UbiComp requerem habilidades de programação avançadas ou não são cuidadosamente projetadas para propósitos educacionais. O estudo foi implementado com 15 professores e 60 estudantes de pós-graduação e graduação de 16 universidades. De acordo com esta pesquisa, os dois conceitos mais desafiadores de UbiComp foram: sensibilidade ao contexto e plataformas de *middleware*. Os resultados também mostraram a dificuldade dos professores em encontrar ferramentas para auxiliar o ensino prático dos conceitos de UbiComp. Dentro desse contexto, este trabalho de mestrado apresenta o design, desenvolvimento e avaliação de um Ambiente de Aprendizagem, chamado Lucy (Learning Ubiquitous Computing Easily), que auxilia as aulas práticas da UbiComp. Lucy tem dois elementos principais: uma ferramenta da Web e um aplicativo para celular Android. O primeiro fornece materiais teóricos de UbiComp, vídeos, práticas e simulações. Já o aplicativo utiliza recursos de *smartphones* e sensores para executar simulações de conceitos de UbiComp de acordo com as configurações feitas na ferramenta Web. Lucy foi avaliada durante as aulas de Sensibilidade ao Contexto em cursos de UbiComp na mesma universidade ao longo de três semestres distintos, nos quais um total de vinte estudantes experimentaram o ambiente. Nas duas primeiras avaliações, reunimos informações sobre as questões pedagógicas de Lucy. Em seguida, realizamos um teste de usabilidade mais formal e uma entrevista semiestruturada sobre o processo de aprendizagem do aluno. Os resultados mostraram melhorias no raciocínio dos alunos sobre os conceitos de UbiComp após as sessões de práticas do Lucy. Além disso, o ambiente ajudou o debate e a interação entre os alunos. O resultado geral do teste de usabilidade foi satisfatório, com pontuação média de 80 pontos de 100 pontos de acordo com a Escala de Usabilidade do Sistema (SUS), que indica um software com boa usabilidade.

Palavras-chave: Computação Ubíqua. Ambiente de Aprendizagem. Web. Dispositivos Móveis. Lucy.

ABSTRACT

Mark Weiser coined the term Ubiquitous Computing (UbiComp) describing a future in which everyday-life objects would have embedded computers providing services anytime and anywhere. One of the hardest activities in teaching UbiComp is putting into practice the theory knowledge of this discipline. In fact, current UbiComp tools require high programming skills or are not carefully designed for educational purposes. We implemented a survey with 15 professors and 60 postgraduate and undergraduate students from 16 universities. According to this survey, the two most challenging UbiComp concepts to explain were: context-awareness and middleware platforms. Results also showed professors' difficulty in finding tools to assist the practical teaching of UbiComp's concepts. Therefore, this work presents the design, development, and evaluation of a Virtual Learning Environment, called Lucy (Learning Ubiquitous Computing Easily), which aids UbiComp practical classes. Lucy has two main elements: a Web tool and an Android mobile app. The former provides UbiComp theoretic materials, videos, practices, and simulations. The latter uses smartphones features and sensors to run simulations of UbiComp concepts. These simulations follow settings configured by the Web tool. We evaluated Lucy during Context-Awareness classes in UbiComp courses, at the same university, along three distinct semesters. We had a total of twenty students that experienced the environment. In the first two evaluations, we gathered information about Lucy's pedagogical issues. Then, we performed a more formal usability test and a semistructured interview concerning the student learning process. Results showed improvements in students reasoning about UbiComp concepts after Lucy practices sessions. Besides, the environment helped the debate and interaction among students. The overall result of the usability test was satisfactory, with the average score of 80 out of 100 points, according to the System Usability Scale (SUS). This result indicates a software with a good usability.

Keywords: Ubiquitous Computing. Learning Environment. Web. Mobile Devices. Lucy.

LISTA DE ILUSTRAÇÕES

Figura 1 – Interfaces de baixa fidelidade do ambiente proposto.	23
Figura 2 – Evolução da Computação Ubíqua.	27
Figura 3 – Definição de contexto segundo (VIANA, 2010).	30
Figura 4 – SenseBoard	32
Figura 5 – Exemplo de aula com LEGO em laboratório.	33
Figura 6 – Visualização geral do simulador de contexto.	34
Figura 7 – Exemplo de informação dos agentes.	34
Figura 8 – Exemplo de aula prática utilizando Phidgets.	36
Figura 9 – Mapa das universidades dos professores pesquisados.	43
Figura 10 – Materiais usados em aulas práticas.	45
Figura 11 – Dificuldades encontradas no ensino de Computação Ubíqua.	46
Figura 12 – Recursos que deveriam existir em um ambiente de ensino para Computação Ubíqua.	46
Figura 13 – Conceitos indispensáveis para um ambiente de ensino para Computação Ubíqua.	46
Figura 14 – Mapa de localização das instituições de ensino dos estudantes pesquisados.	48
Figura 15 – Conceitos considerados difíceis para alunos de pós-graduação da UFC.	49
Figura 16 – Conceitos considerados difíceis para alunos de graduação da UFC.	49
Figura 17 – Recursos necessários para um ambiente de ensino de Computação Ubíqua segundo alunos de pós-graduação da UFC.	50
Figura 18 – Recursos necessários para um ambiente de ensino de Computação Ubíqua segundo alunos de graduação da UFC.	50
Figura 19 – Sugestões e observações de alunos da UFC sobre aulas práticas de Computação Ubíqua.	51
Figura 20 – Conceitos considerados difíceis para alunos de pós-graduação de outras instituições.	51
Figura 21 – Conceitos considerados difíceis para alunos de graduação de outras instituições.	52
Figura 22 – Recursos necessários para um ambiente de ensino de Computação Ubíqua segundo alunos de pós-graduação de outras instituições.	52
Figura 23 – Recursos necessários para um ambiente de ensino de Computação Ubíqua segundo alunos de graduação de outras instituições.	53

Figura 24 – Sugestões e observações de alunos de outras instituições de ensino superior sobre aulas práticas de Computação Ubíqua.	53
Figura 25 – Esquema do funcionamento da simulação de <i>Middleware</i> do Lucy.	57
Figura 26 – (a) Tela de acesso do Lucy. (b) Tela de exibição dos conceitos ubíquos disponíveis. (c) Área interna de um conceito com divisão em <i>Lecture, Simulation</i> e <i>Practice</i>	59
Figura 27 – Diagrama de componentes do Lucy.	59
Figura 28 – Fluxograma para a inserção de um novo conceito no Lucy.	60
Figura 29 – Situação de uso do simulador pelo professor para vários alunos.	62
Figura 30 – Situação de uso do simulador por um aluno para outros alunos.	63
Figura 31 – Situação de uso do simulador por um aluno para observação do resultado em seu próprio <i>smartphone</i>	63
Figura 32 – Exemplificação de contexto para a simulação de Lucy seguindo as definições de Viana (VIANA, 2010).	68
Figura 33 – Estrutura do <i>Context Toolkit framework</i> proposto por Dey.	69
Figura 34 – Arquitetura proposta por Coutaz.	69
Figura 35 – Divisão do conceito de Sensibilidade ao Contexto dentro do Lucy.	70
Figura 36 – Seção de <i>Lecture</i> de Sensibilidade ao Contexto.	70
Figura 37 – Conexão da aplicação móvel com a interface Web de simulação do Lucy. . .	71
Figura 38 – Esquema da tela da simulação sobre Sensibilidade ao Contexto.	72
Figura 39 – Esquema da simulação de Sensibilidade ao Contexto. (a) Camada de sensores. (b) Camada de <i>Widgets</i> . (c) Camada de interpretadores. (d) Camada de agregadores. (e) Camada de <i>Triggers</i>	73
Figura 40 – Ativação das informações dos sensores na camada de <i>widgets</i>	74
Figura 41 – Telas do Lucy móvel. (a) Área inicial dos sensores. (b) Sensores habilitados. (c) Dados dos sensores na camada de <i>widgets</i> . (d) Dados dos sensores na camada de interpretadores.	74
Figura 42 – Reconhecimento de contexto seguro.	75
Figura 43 – Seção de <i>Practice</i> de Sensibilidade ao Contexto.	76
Figura 44 – Diagrama de classes do código Android referente à prática de programação.	77

Figura 45 – Possibilidades diferentes de representação dos valores de luminosidade da primeira caixa da camada de <i>triggers</i> . (a) Valores de luminosidade representados por números inteiros. (b) Valores de luminosidade representados por intervalos (e.g., “ <i>no light</i> ”, “ <i>dim</i> ”, “ <i>low</i> ” e “ <i>bright</i> ”).	79
Figura 46 – Etapas do procedimento realizado nas aulas com o Lucy.	84
Figura 47 – Respostas da avaliação de aspectos pedagógicos dos alunos.	86
Figura 48 – Respostas da avaliação de interface dos alunos.	87
Figura 49 – Respostas da avaliação de aspectos pedagógicos dos professores.	89
Figura 50 – Respostas da avaliação de interface dos professores.	90

LISTA DE QUADROS

Quadro 1 – Comparativo entre os materiais apresentados para as aulas práticas.	39
Quadro 2 – Comparativo entre as propostas de cursos relacionados à Computação Ubíqua.	40
Quadro 3 – Perfil dos professores pesquisados.	44
Quadro 4 – Requisitos funcionais de usuário definidos pelos professores.	47
Quadro 5 – Requisitos funcionais de usuário definidos pelos alunos.	53
Quadro 6 – Comparativo das respostas dos estudantes a uma das questões do questionário do Apêndice C.	85
Quadro 7 – Comparativo entre os materiais apresentados para as aulas práticas incluindo o Lucy.	94
Quadro 8 – Comparativo entre as propostas de cursos relacionados à Computação Ubíqua incluindo o Lucy.	95

LISTA DE ABREVIATURAS E SIGLAS

DCU	<i>Design</i> Centrado no Usuário
FIB	Faculdade Integrada da Bahia
IFBA	Instituto Federal de Educação, Ciência e Tecnologia da Bahia
IFCE	Instituto Federal de Educação, Ciência e Tecnologia do Ceará
Lucy	<i>Learning Ubiquitous Computing Easily</i>
SUS	System Usability Scale
UFAL	Universidade Federal de Alagoas
UFBA	Universidade Federal da Bahia
UFC	Universidade Federal do Ceará
UFCE	Universidade Federal de Campina Grande
UFES	Universidade Federal do Espírito Santo
UFG	Universidade Federal de Goiás
UFPE	Universidade Federal de Pernambuco
UFRN	Universidade Federal do Rio Grande do Norte
UFRPE	Universidade Federal Rural de Pernambuco
UFSM	Universidade Federal de Santa Maria
UFU	Universidade Federal de Uberlândia

SUMÁRIO

1	INTRODUÇÃO	18
1.1	Contextualização	18
1.2	Motivação e Questões de Pesquisa	19
1.3	Objetivos	21
<i>1.3.1</i>	<i>Objetivo geral</i>	21
<i>1.3.2</i>	<i>Objetivos específicos</i>	21
1.4	Contribuições	22
1.5	Metodologia	22
1.6	Organização da Dissertação	24
2	FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS	26
2.1	Computação Ubíqua	26
<i>2.1.1</i>	<i>Conceitos e Características de Computação Ubíqua</i>	28
<i>2.1.2</i>	<i>Sensibilidade ao Contexto</i>	29
2.2	Trabalhos Relacionados	30
<i>2.2.1</i>	<i>Propostas para o ensino de Computação Ubíqua</i>	31
<i>2.2.1.1</i>	<i>Starting with Ubicomp: Using the SenseBoard to Introduce Computing</i>	31
<i>2.2.1.2</i>	<i>Teaching Pervasive Computing to CS Freshmen: A Multidisciplinary Approach</i>	32
<i>2.2.1.3</i>	<i>A Generic Large Scale Simulator for Ubiquitous Computing</i>	33
<i>2.2.1.4</i>	<i>Pervasive Computing as a Classroom-Based Course</i>	35
<i>2.2.1.5</i>	<i>A distance learning curriculum on pervasive computing</i>	36
<i>2.2.1.6</i>	<i>Educating Internet of Things Professionals: The Ambient Intelligence Course</i>	37
<i>2.2.1.7</i>	<i>Análise e discussões</i>	38
2.3	Conclusão	41
3	ENSINO DE COMPUTAÇÃO UBÍQUA	42
3.1	Pesquisa realizada com professores	42
<i>3.1.1</i>	<i>Perfil dos professores</i>	42
<i>3.1.2</i>	<i>Materiais e métodos</i>	43
<i>3.1.3</i>	<i>Resultados</i>	44
<i>3.1.4</i>	<i>Análise e discussões</i>	46
3.2	Pesquisa realizada com estudantes	47

3.2.1	<i>Perfil dos estudantes</i>	47
3.2.2	<i>Materiais e métodos</i>	48
3.2.3	<i>Resultados dos alunos da UFC</i>	49
3.2.4	<i>Resultados dos alunos de diferentes instituições de ensino superior</i>	50
3.2.5	<i>Análise e discussões</i>	52
3.3	Conclusão	53
4	LUCY - LEARNING UBIQUITOUS COMPUTING EASILY	55
4.1	Cenário motivador	55
4.1.1	<i>Uma aula prática de Middleware</i>	55
4.2	Estrutura do Lucy	58
4.2.1	<i>Lecture</i>	60
4.2.2	<i>Simulation</i>	61
4.2.2.1	<i>Situações de utilização do simulador do Lucy</i>	62
4.2.2.2	<i>Página Web de simulação</i>	63
4.2.2.3	<i>Lucy móvel</i>	64
4.2.3	<i>Practice</i>	65
4.3	Conclusão	65
5	PROVA DE CONCEITO	67
5.1	Sensibilidade ao Contexto	67
5.2	Implementação no ambiente Lucy	69
5.2.1	<i>Lecture</i>	69
5.2.2	<i>Simulation</i>	70
5.2.2.1	<i>Interface Web de simulação</i>	71
5.2.2.2	<i>Lucy móvel</i>	72
5.2.2.3	<i>Exemplo de atividade com a simulação</i>	75
5.2.3	<i>Practice</i>	76
5.3	Edição e Customização	78
5.4	Sugestão de uso	79
5.5	Conclusão	80
6	AVALIAÇÃO	81
6.1	Avaliação com estudantes	81
6.1.1	<i>Perfil dos alunos</i>	82

6.1.2	<i>Materiais e métodos</i>	82
6.1.3	<i>Procedimento</i>	83
6.1.4	<i>Resultados</i>	84
6.1.5	<i>Análise e discussões</i>	87
6.2	Avaliação com professores	87
6.2.1	<i>Perfil dos professores</i>	88
6.2.2	<i>Materiais e métodos</i>	88
6.2.3	<i>Procedimentos</i>	88
6.2.4	<i>Resultados</i>	89
6.2.5	<i>Análise e discussões</i>	91
6.3	Conclusão	91
7	CONCLUSÃO	92
7.1	Resultados alcançados	92
7.2	Limitações	95
7.3	Produção bibliográfica	96
7.4	Trabalhos futuros	97
	REFERÊNCIAS	98
	APÊNDICES	101
	APÊNDICE A – Pesquisa com professores sobre aulas de Computação Ubíqua	101
	APÊNDICE B – Levantamento de requisitos para um ambiente interativo de ensino de computação ubíqua - aluno	104
	APÊNDICE C – Questionário pré e pós prática	106
	APÊNDICE D – Questionário guia da atividade de simulação	107
	APÊNDICE E – Avaliação do ambiente lucy	109
	APÊNDICE F – Questionário guia da prática de programação	113
	APÊNDICE G – Avaliação da prática de programação	116
	APÊNDICE H – Avaliação qualitativa - lucy	117

1 INTRODUÇÃO

Esta dissertação de mestrado propõe um ambiente para auxiliar as aulas práticas de Computação Ubíqua. Seus conceitos ensinados em sala de aula são, por vezes, abstratos demais para que os alunos os compreendam somente com aulas teóricas, o que torna as aulas práticas de Computação Ubíqua essenciais para experimentação desses conteúdos. Este capítulo contextualiza esta pesquisa na Seção 1.1. A Seção 1.2 apresenta a motivação e as questões de pesquisa para o desenvolvimento deste trabalho. Na Seção 1.3, estão listados o objetivo geral e os seus desdobramentos específicos. A Seção 1.4 traz as contribuições desta pesquisa. A Seção 1.5 descreve a metodologia utilizada. Por fim, na Seção 1.6, tem-se a organização dos capítulos desta dissertação.

1.1 Contextualização

O progresso tecnológico contribuiu para o barateamento e a disseminação de diversos dispositivos computacionais (e.g. *laptops*, *smartphones*, sensores, atuadores) e também de uma variedade de tecnologias de comunicação sem fio. Esses avanços proporcionaram a concepção e a forte adoção de serviços computacionais embarcados e distribuídos em muitos domínios, tais como domótica, agricultura de alta precisão, transporte urbano, telemedicina 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 do ambiente, misturando-se com objetos do dia a dia (WEISER, 1991). Em “*The Computer for the 21st Century*”, Weiser introduziu o termo Computação Ubíqua. Nesse 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 (WEISER, 1991).

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 (KRUMM, 2010). Ela é um paradigma que envolve várias áreas de Ciência da Computação, compreendendo sistemas distribuídos (BARESI *et al.*, 2006), voláteis (COULOURIS *et al.*, 2011), heterogêneos (SATYANARAYANAN, 2001) e centrados no usuário (WEISER, 1991). 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 dessa área são estudados nas universidades pelo mundo, tais como: captura de experiências e intenções, adaptabilidade, descentralização, descoberta de serviços, heterogeneidade e onipresença (LIMA, 2011). Esses conceitos são abordados tanto em disciplinas de cursos de Ciência da Computação, como em áreas afins. Em geral, o objetivo é que esses conceitos sejam assimilados pelos estudantes e que novos sistemas ubíquos possam ser concebidos e desenvolvidos para auxiliar as pessoas em suas tarefas cotidianas, de forma que esses 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 essas ferramentas são muito escassas, fugazes e, às vezes, de difícil acesso. Os altos custos do *hardware* utilizado, como por exemplo o LEGO Mindstorms (SILVIS-CIVIDJIAN, 2015), caracteriza também uma das dificuldades encontradas.

É nesse contexto que esta pesquisa busca apoiar o ensino-aprendizagem de conceitos de Computação Ubíqua por meio da concepção, desenvolvimento e avaliação de um ambiente 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. Esse ambiente consiste de duas partes: uma aplicação Android, que acessa os sensores do dispositivo e apresenta o resultado de simulações configuradas; e uma aplicação Web, que conta com conteúdos teóricos, configuração de simulações e práticas de programação para cada conceito estudado.

1.2 Motivação e Questões de Pesquisa

Conceitos de Computação Ubíqua são vistos em diferentes disciplinas nas universidades. Podem-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 (CHALMERS, 2015), por exemplo, é descrito 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 (CHALMERS, 2015).

Embora existam na literatura diversos relatos de experiências no ensino de conceitos de Computação Ubíqua, tais como (RICHARDS *et al.*, 2012), (SILVIS-CIVIDJIAN, 2015) e (CHALMERS, 2015), 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-RED¹, voltado para Internet das Coisas, e o App Inventor², voltado para o ensino de lógica de programação, ou mesmo ambientes de programação tradicionais de aplicações móveis (e.g., Android Studio³).

Segundo (CHALMERS, 2015), em algumas universidades, o interesse em determinadas plataformas e em linguagens de programação, assim como o conhecimento prévio dos estudantes, podem levar os professores a diferentes escolhas para a utilização de artefatos que apoiem as aulas práticas. Por exemplo, o professor pode optar pelo uso do Arduino⁴ 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.

Mesmo com a demanda por ambientes ou ferramentas para auxiliar a prática de conceitos ubíquos, há uma escassez significativa desses recursos. Com isso, componentes com outras finalidades base, como por exemplo Raspberry, Beaglebone e Arduino, são utilizados para auxiliar as aulas de Computação Ubíqua, o que pode exigir mais empenho e criatividade dos professores para a elaboração de atividades práticas.

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 (DUARTE, 2014). Levando em consideração (HAMDAOUI *et al.*, 2013), 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

¹ <https://nodered.org/>

² <http://appinventor.mit.edu>

³ <https://developer.android.com/studio/index.html?hl=pt-br>

⁴ <https://www.arduino.cc/>

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

Este trabalho foi norteado por duas questões de pesquisa:

- **Questão 1 - (QP1)** - Como ocorre o ensino de Computação Ubíqua nas universidades?
- **Questão 2 - (QP2)** - Quais características devem existir em um ambiente de apoio ao ensino de Computação Ubíqua de forma a favorecer a compreensão dos estudantes acerca de conceitos teóricos dessa área?

1.3 Objetivos

Esta seção discute o objetivo geral e os objetivos específicos desta pesquisa com base nas questões levantadas na Seção 1.2.

1.3.1 *Objetivo geral*

Caracteriza-se como objetivo geral deste trabalho a proposta de um ambiente simples e economicamente viável para a execução de aulas práticas com a finalidade de favorecer o ensino dos conceitos de Computação Ubíqua.

1.3.2 *Objetivos específicos*

- Entender como funciona o ensino de Computação Ubíqua nas universidades, levando em consideração a organização das aulas, suas práticas e materiais didáticos. Verificou-se, com alunos e professores, quais são as maiores dificuldades, quais são os assuntos de Computação Ubíqua considerados mais importantes, como é a organização das aulas, se possuíam aulas práticas e quais os materiais utilizados no decorrer dos cursos investigados.
- Conceber e desenvolver um ambiente com suporte a dispositivos móveis que possa auxiliar nas aulas práticas de Computação Ubíqua.
- Avaliar os resultados do uso do ambiente em sala de aula junto a alunos e professores.

1.4 Contribuições

A fim de alcançar os objetivos propostos e com base no contexto exposto, foi criado um ambiente nomeado *Learning Ubiquitous Computing Easily* (Lucy). Ele possui uma parte Web e outra móvel para dispositivos Android. As duas partes se conectam, conforme ilustra a Figura 1, para que as configurações feitas no lado Web se reflitam na aplicação móvel. Esse ambiente pode auxiliar professores e alunos na prática de conceitos teóricos de Computação Ubíqua. Atualmente, essa ferramenta foca em sensibilidade ao contexto, mas também possui a capacidade de abordar outros temas ubíquos, tais como adaptabilidade, descoberta automática de serviços, heterogeneidade, interoperabilidade e *middleware*. Para cada conceito, são disponibilizadas seções com informações teóricas, exemplos de implementações, simulações das camadas de um sistema ubíquo e práticas de programação relacionadas a cada assunto estudado.

A ideia é que para cada conceito estudado (e.g., heterogeneidade e sensibilidade ao contexto) sejam criadas estruturas de simulações que exemplifiquem o funcionamento desses conceitos. Os alunos podem visualizar 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.

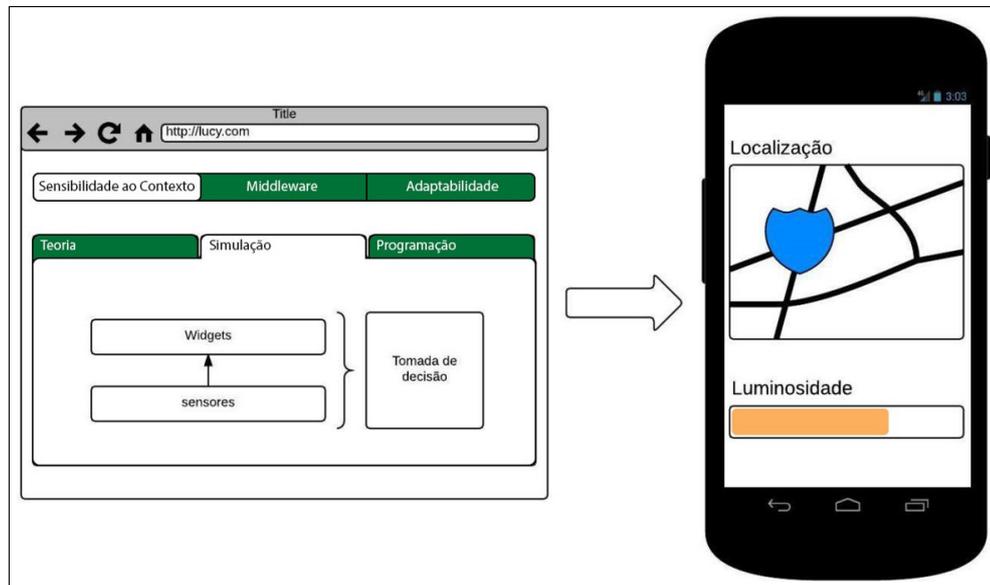
Espera-se favorecer a compreensão dos estudantes acerca dos conceitos de Computação Ubíqua, visando aumentar a criação de aplicações ubíquas e difundir a utilização de mais dispositivos, que se conectarão a outros dispositivos, pessoas, coisas e lugares.

Com base nas estruturas de cursos de Computação Ubíqua investigadas ao longo deste trabalho, espera-se também fornecer subsídios para a organização de disciplinas de Computação Ubíqua. Neste modelo, são propostas organizações de aulas com a utilização do ambiente Lucy, que deverá auxiliar aulas teóricas e principalmente aulas práticas, dispondo de estruturas específicas para realização de simulações e práticas de programação.

1.5 Metodologia

A metodologia utilizada neste trabalho se baseou no processo de *Design Centrado no Usuário* (DCU), que possui sua base formal na norma ABNT NBR ISO 9241-210:2011 (Ergonomia da interação humano-sistema - Parte 210: Projeto centrado no ser humano para sistemas interativos) (ABNT, 2011). No DCU, o desenvolvedor não só imagina como será a interação do usuário final com o sistema, produto ou serviço em construção, mas também realiza

Figura 1 – Interfaces de baixa fidelidade do ambiente proposto.



Fonte – o autor.

uma série de testes com usuários reais para dar validade ao trabalho.

Segundo (WOODSON *et al.*, 1981), 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 (WOOD; ROMERO, 2010).

A primeira e constante etapa desta pesquisa foi investigar o cenário dos cursos de Computação Ubíqua nas universidades. O objetivo foi identificar a estrutura das aulas, principais conceitos estudados e quais as maiores dificuldades de professores e alunos. Essa etapa foi realizada por meio de estudos bibliográficos e questionários aplicados a sessenta alunos e quinze professores de diversas universidades. Esse momento de investigação e estudo do estado da arte se estendeu durante todo o desenvolvimento deste trabalho.

Uma vez identificada a necessidade de um ambiente de apoio às aulas práticas de Computação Ubíqua, o DCU foi escolhido para validar esta pesquisa. Para evitar excesso de informações e funcionalidades inúteis, além de deixar essa ferramenta de acordo com as necessidades reais dos usuários finais, as quatro etapas principais do DCU foram seguidas:

- **Especificação do contexto de uso:** essa etapa orientou a definição dos usuários do ambiente, para qual finalidade este seria utilizado e de que forma se daria essa utilização. Essas determinações foram feitas com base nas respostas obtidas

na fase de investigação e estudo do estado da arte do cenário dos cursos de Computação Ubíqua.

- **Especificação dos requisitos dos usuários:** Nesta fase, foi necessário identificar os requisitos de negócio e os objetivos dos usuários a fim de se construir um ambiente que atendesse às necessidades especificadas. Dessa forma, procurou-se conhecer: os conceitos ubíquos considerados mais importantes por professores; os conceitos ubíquos considerados mais difíceis de se entender na ausência de aulas práticas segundo os estudantes; e as características e funcionalidades que um ambiente de apoio às aulas práticas de Computação Ubíqua deveria ter na opinião desses usuários.
- **Criação de soluções de projeto:** Nesta fase, protótipos do ambiente foram criados levando sempre em consideração a opinião dos usuários para o refinamento e aprimoramento do produto. A proposta da versão final do Lucy passou a ser delimitada conforme cada teste realizado. A partir dos resultados dos testes, os pedidos de melhoria de características e funcionalidades apresentados eram implementados.
- **Avaliação do projeto:** Nesta etapa de avaliações, alunos e professores foram convidados a usar o Lucy. O objetivo era avaliar a usabilidade, a interface, a comunicação visual e o potencial pedagógico do sistema. Em momentos com atividades direcionadas e outras livres, os usuários utilizaram a ferramenta e forneceram *feedback* dos pontos fortes e fracos da solução proposta. Questionários avaliativos relacionados ao ambiente também foram aplicados com o intuito de verificar se os requisitos de estudantes e professores foram atingidos.

1.6 Organização da Dissertação

Esta dissertação está organizada em oito capítulos, sendo este primeiro responsável por contextualizar o assunto abordado por este trabalho e apresentar a motivação, os objetivos, as contribuições e a metodologia. Os demais capítulos estão organizados da seguinte forma:

- O Capítulo 2 apresenta a fundamentação teórica em que são explorados os conceitos teóricos relacionados aos temas desta pesquisa, tais como as definições e características de Computação Ubíqua, e mostra ainda os trabalhos relacionados, os quais trazem exemplos de materiais usados para aulas práticas de Computação Ubíqua, assim como organização e

estruturas de cursos relacionados a essa área.

- O Capítulo 3 traz uma pesquisa realizada com professores e alunos sobre o ensino de Computação Ubíqua no Brasil e fora dele.
- O Capítulo 4 descreve um ambiente criado especialmente para auxiliar as aulas práticas de Computação Ubíqua.
- O Capítulo 5 apresenta o conceito de Sensibilidade ao Contexto, criado dentro do ambiente Lucy, com seções de teoria, simulação e prática como prova de conceito.
- O Capítulo 6 descreve as avaliações feitas com professores e alunos em testes de usabilidade, interface, aspectos pedagógicos e satisfação com o Lucy. Apresenta também os resultados destas avaliações realizadas.
- Por fim, o Capítulo 7 mostra a conclusão e os trabalhos futuros relacionados a esta pesquisa.

2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

Neste capítulo, são apresentados os principais conceitos teóricos relacionados a esta pesquisa. Na Seção 2.1, é exposta a origem da Computação Ubíqua, sua definição e evolução, além da descrição dos principais conceitos e características. A Seção 2.2 traz exemplos de recursos utilizados para a realização de aulas práticas de Computação Ubíqua e descreve estrutura e organização curricular de cursos ligados a esse tema. Por fim, na Seção 2.3, são mostradas as conclusões sobre os temas, trabalhos apresentados e comparações feitas entre eles e detalhados neste capítulo.

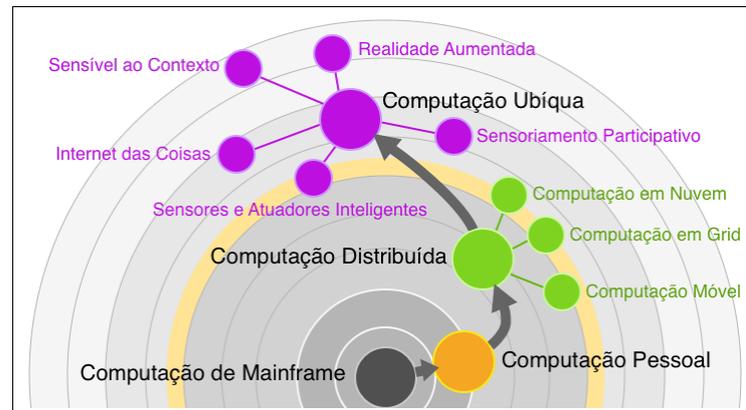
2.1 Computação Ubíqua

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 relacionadas ao trabalho como em atividades lúdicas (KRUMM, 2010). 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 (WEISER, 1991).

A realização da Computação Ubíqua tornou-se possível graças à evolução do *hardware* (e.g., *smartphones*, *tablets*, sensores e outros equipamentos com alto poder de comunicação e processamento) e de *software* (FILHO, 2012). A Computação Ubíqua pode ser vista como fruto da evolução de outros paradigmas que tiveram início com a computação de *mainframe*. A Figura 2 ilustra essa afirmação. Em uma primeira fase da computação, 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 funcionam como se fossem um único sistema. Então, na quarta fase ou fase atual, há a concretização da Computação Ubíqua (BOYCE; NEALE, 2006).

Existem vários conceitos que caracterizam e compõem a Computação Ubíqua. Eles

Figura 2 – Evolução da Computação Ubíqua.



Fonte – Adaptado de (KNAPPEYER *et al.*, 2013)

possuem foco diversificado, mas se assemelham no fato de dispersar suas atuações pelo ambiente onde estão inseridos. A Figura 2 apresenta algumas ramificações que exemplificam alguns desses conceitos integrantes de Computação Ubíqua, a qual destaca a interação de seres humanos com recursos computacionais distribuídos pervasivamente pelo ambiente (KNAPPEYER *et al.*, 2013).

Segundo (NIEUWDORP, 2007), Computação Pervasiva se refere à capacidade de acessar serviços e informações a qualquer hora e em qualquer lugar. No entanto, há incertezas consideráveis que inibem uma definição geral do termo pervasivo no discurso computacional. Isso acontece, principalmente, devido à IBM, que cunhou o termo em 1998, apresentar uma definição de Computação Pervasiva que é extremamente ambígua, resultando em várias explicações para o mesmo termo na mesma empresa (NIEUWDORP, 2007).

Uma das diferenças que pode ser apontada entre a Computação Ubíqua e a Computação Pervasiva é a manutenção da conexão e da configuração de serviços necessários mesmo durante a mobilidade do usuário. Isso significa dizer que a Computação Ubíqua se preocupa em manter a conexão e garantir que os serviços estejam configurados conforme as definições e necessidades de seus usuários apesar de sua movimentação. O conceito de pervasividade diz respeito à embarcação de computadores no ambiente de forma invisível ao usuário e sem garantia de muita mobilidade (ARAUJO, 2003). Apesar dessas diferenças, a definição dos dois conceitos se aproxima bastante, o que leva este trabalho a considerar Computação Ubíqua e Pervasiva como sinônimas.

2.1.1 *Conceitos e Características de Computação Ubíqua*

Com base em (LIMA, 2011) e em (FILHO, 2012), após observados alguns projetos de sistemas ubíquos, foram levantadas características fundamentais para a descrição dos conceitos proeminentes de Computação Ubíqua. Essas características são, segundo (LIMA, 2011; FILHO, 2012):

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, os quais 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 e alterar sua atuação conforme essas necessidades;
10. **Tolerância a falhas:** capacidade do sistema se recuperar após falhas sem deixar que o usuário perceba.

A Computação Ubíqua concretiza-se por meio de sistemas computacionais que possuem componentes de *hardware* e de *software* integrados para a realização de tarefas e que executam de forma coordenada as atividades, que incluem troca de informações, invocações de serviços e compartilhamento de recursos (LIMA, 2011). Hoje, 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.

2.1.2 *Sensibilidade ao Contexto*

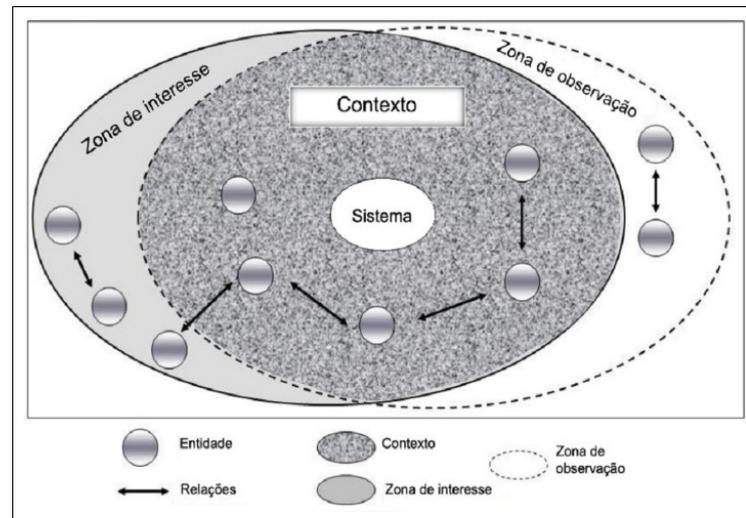
Segundo a definição de Dey (DEY, 2001), contexto é qualquer informação que pode ser usada para caracterizar a situação de qualquer entidade. Entende-se como entidade uma pessoa, lugar ou objeto considerado relevante para a interação entre um usuário e uma aplicação, incluindo o próprio usuário e a própria aplicação. Dito isso, um sistema sensível ao contexto, segundo a definição de Dey (DEY, 2001), é aquele que se utiliza do contexto para fornecer informações e/ ou serviços relevantes ao usuário, sendo essa relevância dependente do que o usuário deseja fazer.

Para (COUTAZ *et al.*, 2005), contexto é um conjunto de informações definidas por entidades (sendo estas objetos ou mesmo informações), funções ou papéis exercidos pelas entidades, as relações presentes entre essas entidades e as situações em que elas estão inseridas. O contexto então faz parte do processo de interação com um ambiente em constante mudança em que sistemas podem operar (COUTAZ *et al.*, 2005).

Complementando a definição de Dey e de Coutaz, (VIANA, 2010) define contexto como um ponto de interseção entre a zona de observação e a zona de interesse de um sistema ubíquo. Define-se como zona de observação tudo aquilo que pode ser observado em um determinado instante (e.g., coordenadas e temperatura de um local). Zona de interesse é o conjunto de entidades relevantes para o sistema (e.g., temperatura de um local para um sistema de alarme de incêndios). Esse ponto de interseção é dinâmico e evolutivo, isso porque o sistema pode modificar seu interesse ao longo da execução de uma tarefa e as informações observadas dependem dos elementos de captura acessíveis em um dado instante. Então, a zona de contexto passa a ser esse ponto comum entre a zona de observação e a zona de interesse, como mostra a Figura 3.

O reconhecimento das informações contextuais são indispensáveis para os mais diversos tipos de sistemas e aplicações que desejam fornecer serviços e/ ou informações personalizados a seus usuários. Como exemplo de personalização de serviços e diversificação, pode-se citar um *middleware* de contexto que integra sistemas de gestão de eventos ligados à transmissão de energia elétrica. Essa integração auxilia nas tomadas de decisão necessárias à intervenção

Figura 3 – Definição de contexto segundo (VIANA, 2010).



Fonte – (DUARTE *et al.*, 2015)

operativa para prover transmissão de energia. As tomadas de decisão são baseadas no que o sistema fornece como informações relevantes para o usuário, sendo essa relevância definida pelo seu contexto (QUINTINO; FERRAZ, 2014).

2.2 Trabalhos Relacionados

Nesta seção, são apresentados alguns trabalhos encontrados que estão diretamente relacionados com esta pesquisa. Foram incluídos materiais e ferramentas utilizados no ensino de Computação Ubíqua, além de trabalhos com a organização curricular de cursos de Computação Ubíqua.

Os exemplos aqui relatados como trabalhos relacionados foram extraídos de uma pesquisa bibliográfica mais ampla, feita nas bases do Google Scholar¹, Springer², IEEE Xplore³ e ACM Digital Library⁴, com as seguintes palavras-chave: “*computing teaching environment*”; “*ubiquitous computing teaching environment*”; “*ubiquitous computing software*”; “*teaching environment for context awareness*”; “*teaching ubiquitous computing*”; “*context awareness teaching*”.

As pesquisas com as palavras-chave mais específicas sobre ambientes ou *softwares* para o ensino de Computação Ubíqua não trouxeram resultados de outros trabalhos semelhantes

¹ <https://scholar.google.com.br/>

² <http://www.springer.com/br/>

³ <http://ieeexplore.ieee.org/Xplore/home.jsp>

⁴ <http://dl.acm.org/>

ao que é proposto pelo Lucy. Diante disso, foram consideradas propostas mais gerais de aulas, metodologias e materiais utilizados para o ensino de conceitos de Computação Ubíqua. Além disso, a busca passou a ser mais delimitada ao conceito de Sensibilidade ao Contexto, com termos como “*context awareness teaching*”, por exemplo. Alguns trabalhos oriundos dessa busca estão descritos a seguir.

2.2.1 *Propostas para o ensino de Computação Ubíqua*

Nas pesquisas realizadas, o mais comum foi encontrar trabalhos que se utilizam de Computação Ubíqua para auxiliar o ensino das mais diversas áreas. Há uma maior dificuldade de se encontrar trabalhos que apresentem o ensino prático de Computação Ubíqua com a utilização de alguma tecnologia. Ainda assim, foram encontrados alguns estudos que descrevem materiais, métodos e ferramentas utilizados em atividades práticas de conceitos ubíquos. A seguir, estão discriminados alguns desses trabalhos.

2.2.1.1 *Starting with Ubicomp: Using the SenseBoard to Introduce Computing*

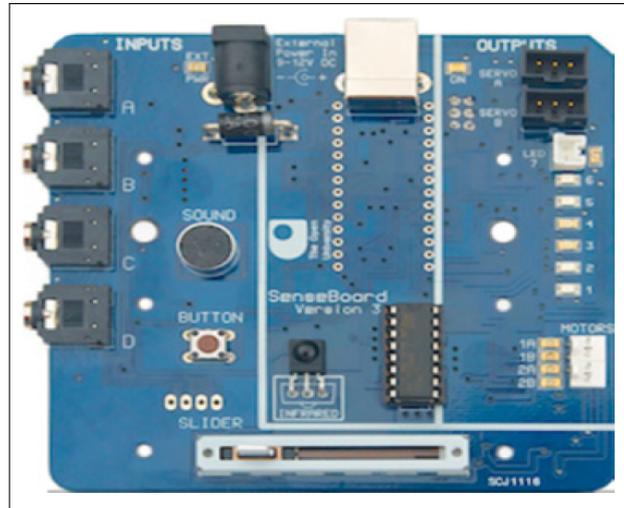
Em (RICHARDS *et al.*, 2012), tem-se a descrição de um módulo, *My Digital Life* (TU100), criado para auxiliar estudantes novatos da graduação da educação a distância no desenvolvimento de programas ubíquos. Eles experimentam, de maneira lúdica, a prática de conceitos de Computação Ubíqua por meio de um *hardware* chamado de SenseBoard, que pode ser programado com a linguagem Sense, uma linguagem em blocos semelhante ao Scratch⁵. O dispositivo SenseBoard, citado em (RICHARDS *et al.*, 2012), é uma placa, com *input*, *output* e alguns sensores, baseado no microcontrolador Arduino, conforme ilustra a Figura 4.

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 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 a distância por alunos novatos da *Open University*.

Como o público-alvo de (RICHARDS *et al.*, 2012) 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

⁵ <https://scratch.mit.edu/projects/editor/>

Figura 4 – SenseBoard



Fonte – (RICHARDS *et al.*, 2012)

ligação de blocos.

É possível observar que (RICHARDS *et al.*, 2012) trabalha o ensino de Computação Ubíqua por meio da experimentação prática sem exigências de conhecimentos prévios em programação. Os conceitos desse tema desenvolvidos nas atividades práticas estão mais relacionados ao funcionamento de estruturas baseadas em localização com o uso de sensores e serviços. 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.

2.2.1.2 *Teaching Pervasive Computing to CS Freshmen: A Multidisciplinary Approach*

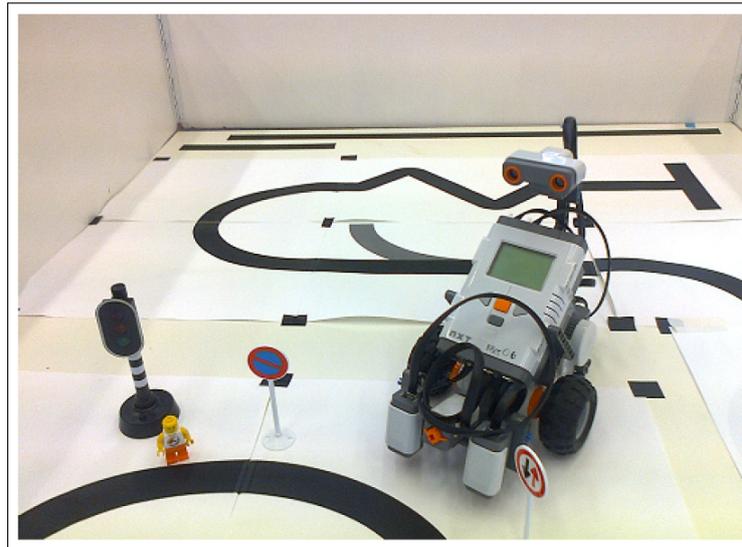
Em (SILVIS-CIVIDJIAN, 2015), há a preocupação de ensinar Computação Pervasiva⁶ a estudantes do primeiro ano do curso de Ciência da Computação. A ideia desse trabalho é que a aprendizagem seja por meio do desenvolvimento de projetos pervasivos, que possam inferir o contexto dos alunos através do reconhecimento de padrões, por exemplo. Os alunos são desafiados com a metodologia da aprendizagem baseada em problemas a encontrar soluções que melhorem a qualidade de vida das pessoas. Assim, podem construir cadeiras de rodas ou carros inteligentes que tomem decisões a partir do reconhecimento de padrões das atividades cerebrais do utilizador.

Além de projetar e desenvolver os sistemas pervasivos, os alunos são levados a

⁶ O termo Computação Pervasiva é sinônimo, nesta pesquisa, de Computação Ubíqua.

demonstrar as funcionalidades de seus projetos e como os fizeram. Para fornecer suporte a esses projetos práticos, como ilustra a Figura 5, existem laboratórios equipados com LEGO Mindstorms NXT 2.0. Esse material (LEGO) é usado na construção da parte física do projeto a ser controlada por meio de programas e algoritmos elaborados no ambiente matemático MATLAB.

Figura 5 – Exemplo de aula com LEGO em laboratório.



Fonte – (SILVIS-CIVIDJIAN, 2015)

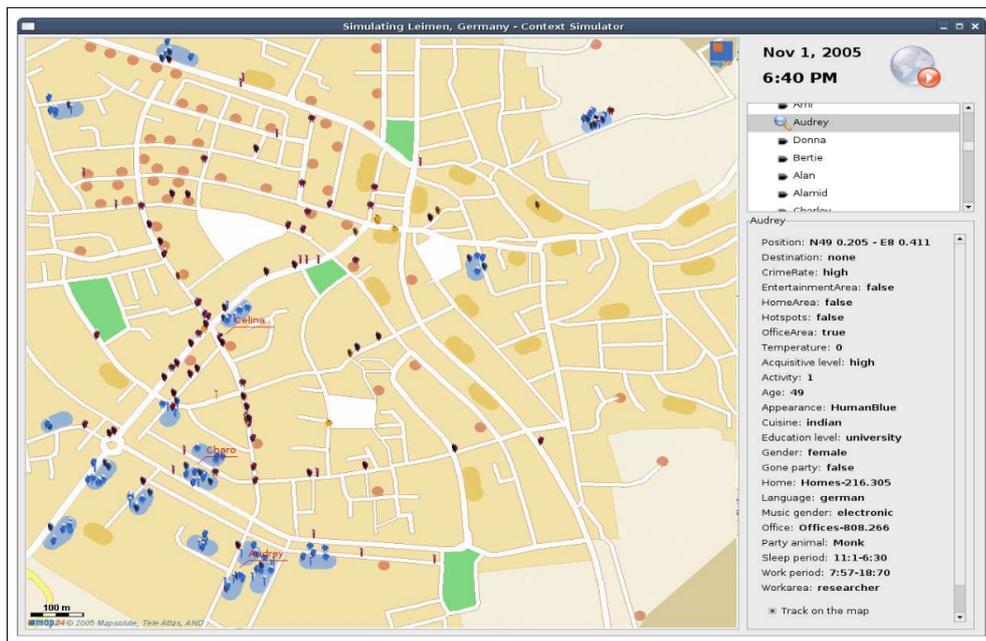
Apesar de ser um *hardware* robusto, fácil de programar, móvel e escalável, os pontos negativos de propostas de ensino com LEGO são que ele possui custo alto e limitação no que diz respeito aos seus tipos de sensores e atuadores. Com isso, existe também uma forte limitação da variedade de sistemas ou projetos pervasivos que podem ser construídos.

2.2.1.3 A Generic Large Scale Simulator for Ubiquitous Computing

O trabalho de (MARTIN; NURMI, 2006) se constitui de um simulador que exemplifica a execução de regras contextuais criadas para um ambiente específico com diversos agentes diferenciados e únicos. A Figura 6 mostra uma visão generalizada do simulador em execução. Na coluna do lado direito dessa figura, é possível observar a data e a passagem do tempo conforme a execução do simulador, além de informações do contexto de cada agente presente na simulação. A Figura 7 apresenta quais informações podem ser observadas de cada agente, do qual então se tem a posição atual e as informações relacionadas a essa localização, por exemplo: se é o destino do agente, se esse local é perigoso ou se é seu local de trabalho ou moradia.

Segundo as definições da norma ISO/IEC/IEEE 24765:2010 (ISO, 2011), simulação se caracteriza por tudo aquilo que se comporta ou funciona como um determinado sistema quando fornecido um conjunto de parâmetros. É também algo que representa características comportamentais selecionadas de um sistema físico ou abstrato. Por essa definição, simulador é qualquer dispositivo, programa de computador ou sistema que realiza uma simulação.

Figura 6 – Visualização geral do simulador de contexto.



Fonte – (MARTIN; NURMI, 2006)

Figura 7 – Exemplo de informação dos agentes.



Fonte – (MARTIN; NURMI, 2006)

O simulador desse trabalho faz parte de um conjunto de ferramentas chamado Siafu⁷ escrito na linguagem Java. Esse conjunto de ferramentas não foi criado visando o ensino-aprendizagem do conceito ubíquo do qual ele trata, Sensibilidade ao Contexto. No entanto, ele

⁷ <http://siafusimulator.org/>

possui exemplos de simulações prontas para serem executadas e visualizadas, trazendo também a possibilidade de extensão do código Java para a criação de outros ambientes, agentes e regras. Para a criação de novas simulações, três classes do código original precisam ser estendidas e definidas: *BaseAgentModel*, *BaseWorldModel* e *BaseContextModel*.

Um professor poderia utilizar os exemplos já prontos desse conjunto de ferramentas para ilustrar em sala de aula o que pode ser considerado contexto, os elementos que compõem o contexto e o que acontece diante da variação desse contexto. Apesar disso, haveria restrição para o uso desse material no trabalho de outros conceitos ubíquos, pois o simulador em questão se restringe à Sensibilidade ao Contexto.

2.2.1.4 *Pervasive Computing as a Classroom-Based Course*

Em (CHALMERS, 2015), é descrito um cenário de organização de disciplinas que possuem conceitos e práticas de Computação Ubíqua na Universidade de Sussex, no Reino Unido. Essas aulas são para estudantes de graduação e pós-graduação e estão organizadas em três partes: exposição teórica, seminários e práticas em laboratório (CHALMERS, 2015).

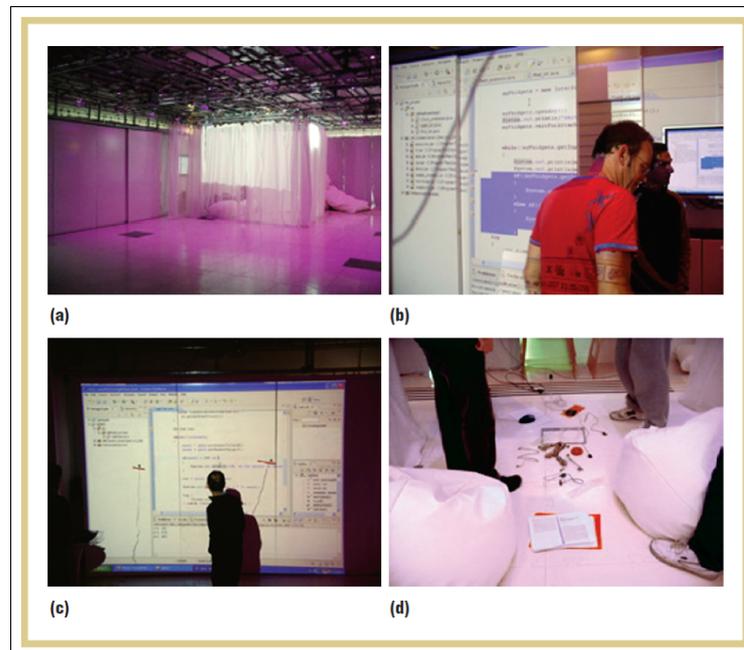
Na exposição teórica, são apresentados os conceitos fundamentais de Computação Ubíqua e trabalhos que abrangem, de forma mais generalizada, o estado da arte relacionado ao tema. No momento da apresentação de seminários, o foco é analisar criticamente as teorias e as soluções existentes para resolução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

Nas práticas em laboratório, o objetivo é programar variados dispositivos para exercitar 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. As aulas práticas em laboratório duram em média duas horas.

Dispositivos chamados Phidgets são usados nas aulas práticas, conforme ilustra a Figura 8. 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. Segundo o autor, essa escolha decorre do fato de que o uso de Motes ou Arduinos causaria uma mudança no foco da

Figura 8 – Exemplo de aula prática utilizando Phidgets.



Fonte – (CHALMERS, 2015)

disciplina, que passaria a requerer dos estudantes maior conhecimento de programação em C e programação de *hardwares* embarcados.

O ponto negativo desse trabalho é a ausência de um ambiente de apoio às aulas práticas para auxiliar na visualização e assimilação dos conceitos apresentados. Não há uma ferramenta que concentre atividades teóricas e práticas em um mesmo espaço com o objetivo de relacioná-las diretamente.

2.2.1.5 *A distance learning curriculum on pervasive computing*

O trabalho desenvolvido em (GOUMOPOULOS *et al.*, 2017) apresenta um currículo de um curso de Computação Ubíqua criado para a pós-graduação a distância da *Hellenic Open University*, na Grécia. Por ser um curso a distância, este currículo possui apenas 5 encontros presenciais a cada ano letivo. Nesses encontros, são discutidos problemas teóricos e conceitos relacionados à Computação Ubíqua.

Esse curso também divide suas atividades em teorias e práticas. As atividades teóricas incluem estudos sobre a literatura científica considerada relevante para o curso, apresentações e discussões de trabalhos relacionados aos tópicos estudados. Os encontros presenciais são responsáveis por parte da realização dessas atividades teóricas, pois neles também ocorrem exercícios sobre os temas relacionados à Computação Ubíqua e apresentações de trabalhos.

As atividades práticas envolvem o desenvolvimento de aplicações, com Android por exemplo, após demonstração de estudos de casos que explicam o passo a passo do que precisa ser feito. Várias tecnologias também são utilizadas nesses exercícios práticos, tais como LEGO Mindstorms, Phidgets e Arduino. Além disso, tem-se a utilização de *softwares* como o 123d circuits⁸ e o Wireshark⁹. Há ainda o suporte de um laboratório remoto de Arduino para a realização dos exercícios práticos: <http://pc2.dsmc.eap.gr/>.

A proposta de (GOUMOPOULOS *et al.*, 2017) engloba o emprego de variadas tecnologias na realização de atividades teóricas e práticas. Por ser um curso a distância, a utilização de um ambiente para concentrar as operações do processo de ensino-aprendizagem é indispensável, mas o trabalho não relaciona as ligações dos exercícios teóricos com os práticos dentro desse ambiente.

2.2.1.6 *Educating Internet of Things Professionals: The Ambient Intelligence Course*

O trabalho de (CORNO *et al.*, 2016) apresenta um curso que, apesar de não ser direta e especificamente sobre Computação Ubíqua, trabalha inúmeros conceitos relacionados a esta área em seus projetos e definições de ambientes inteligentes.

Inteligência Ambiental é um curso voltado para a graduação de *Politecnico di Torino*, na Itália. Esse curso foca no desenvolvimento de sistemas para ambientes inteligentes por meio do trabalho em equipe. As equipes são constituídas de estudantes de diferentes programas acadêmicos, tais como arquitetura, engenharia de computação, engenharia elétrica, engenharia mecânica e *design* industrial. Assim, cada membro da equipe tem uma função determinada com base em seus conhecimentos específicos para a construção de sistemas para ambientes inteligentes.

Esse curso está dividido em quatro seções de atividades de diferentes tipos. 35% do curso está reservado para aulas teóricas. Atividades práticas e exercícios em sala de aula que envolvem a definição de soluções para problemas propostos englobam 30% do curso. Exercícios guiados no laboratório conforme os tópicos estudados compreendem um total de 15%. Por fim, trabalhos em grupos supervisionados no laboratório abrangem 20% do curso. Alguns *softwares* e *hardwares* são utilizados ainda como apoio às atividades, tais como Raspberry Pi, Android, protocolos Web e sistemas *Smart home*.

⁸ <https://123d.circuits.io/>

⁹ <https://www.wireshark.org/>

O curso apresentado por (CORNO *et al.*, 2016) aborda diversos conceitos de Computação Ubíqua e possui uma estrutura bem definida para momentos teóricos e práticos, totalizando uma carga horária maior para as atividades práticas, o que equivale a 65% de todo o curso. A proposta de desenvolvimento de projetos com base em resolução de problemas por equipes, utilizada nesse curso, também poderia se aplicar a estruturas de outros programas voltados ao ensino-aprendizagem de conceitos de Computação Ubíqua.

2.2.1.7 *Análise e discussões*

Os recursos apresentados como alternativas para apoiar as aulas práticas de Computação Ubíqua são materiais limitados de certa forma, pois ou não foram projetados especificamente para esse fim ou foram descontinuados, o que é o caso do trabalho apresentado por (RICHARDS *et al.*, 2012), que parou de ser comercializado e difundido. Uma das principais vantagens destacadas para essas ferramentas é que elas permitem o desenvolvimento de aulas práticas para se trabalhar múltiplos conceitos ubíquos, com exceção da proposta descrita em (MARTIN; NURMI, 2006), que se restringe ao conceito de Sensibilidade ao Contexto. No Quadro 1, há um comparativo entre esses trabalhos, resumindo suas características relevantes para esta pesquisa.

Nos modelos de organização de cursos apresentados há similaridades no que se refere à estrutura das aulas, que se dividem em teóricas e práticas. Os materiais utilizados em cada modelo de curso varia conforme os objetivos de cada um. Verificou-se que estruturas semelhantes de aulas podem ser aplicadas tanto a alunos de graduação como aos de pós-graduação independente da universidade. Um quadro comparativo das propostas de cursos relacionados à Computação Ubíqua pode ser vista no Quadro 2.

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. Estas são fatores essenciais em cursos relacionados à Computação Ubíqua, pois favorecem uma aprendizagem mais ativa, transformando os alunos em agentes atuantes no processo de ensino-aprendizagem. Estruturas de aulas práticas realizadas pelos trabalhos apresentados nesta seção estão descritas na última coluna do Quadro 2.

Quadro 1 – Comparativo entre os materiais apresentados para as aulas práticas.

	Trabalha diferentes conceitos de Computação Ubíqua	Nível de Conhecimento de programação exigido	Vantagens	Desvantagens
SenseBoard (RICHARDS <i>et al.</i> , 2012)	Sim	Sem exigências	<ul style="list-style-type: none"> - Permite trabalhar diversos conceitos de Computação Ubíqua. - Não exige conhecimentos prévios em programação. 	<ul style="list-style-type: none"> - O hardware SenseBoard não está mais disponível. - A linguagem em blocos Sense está instável.
LEGO (SILVICIVIDJIAN, 2015)	Sim	Sem exigências	<ul style="list-style-type: none"> - Permite trabalhar diversos conceitos de Computação Ubíqua. - Não exige conhecimentos prévios em programação. 	<ul style="list-style-type: none"> - Custo médio/alto para aquisição e manutenção. - Limitação dos tipos de sensores e atuadores utilizados.
Phidgets (CHALMERS, 2015)	Sim	Conhecimentos de programação	<ul style="list-style-type: none"> - Permite trabalhar diversos conceitos de Computação Ubíqua. - Fornece APIs em várias linguagens de programação. 	<ul style="list-style-type: none"> - Custo médio/alto para aquisição. - Não há um ambiente/sistema que relacione aulas teóricas e práticas.
Simulador de sistemas sensíveis ao contexto (MARTIN; NURMI, 2006)	Não – Apenas sensibilidade ao contexto	Conhecimentos de programação em Java	<ul style="list-style-type: none"> - Permite a visualização ilustrada do conceito estudado. - Possibilita ao aluno a criação própria de uma simulação. 	<ul style="list-style-type: none"> - Não foi criado para o ensino-aprendizagem de conceitos ubíquos. - Não há interação com os agentes na interface de simulação criada.

Fonte – o autor.

Quadro 2 – Comparativo entre as propostas de cursos relacionados à Computação Ubíqua.

Curso de Computação Ubíqua	Local	Público-Alvo	Divisão das aulas/atividades	Recursos usados	Estrutura das aulas práticas
(CHALMERS, 2015)	Universidade de Sussex, no Reino Unido	Estudantes de graduação e pós-graduação	Exposição teórica, seminários e práticas em laboratório	Phidgets	Desenvolvimento de experimentos direcionados usando Phidgets.
Curso de Computação Ubíqua a distância (GOU-MOPOULOS <i>et al.</i> , 2017)	Hellenic Open University, na Grécia	Estudantes de pós-graduação a distância	Atividades teóricas e práticas	Android, LEGO Mindstorms, Phidgets, Arduino, 123d circuits, WireShark e laboratório remoto de Arduino	Desenvolvimento de aplicações direcionadas com as tecnologias citadas e resolução de exercícios.
Curso de Intelligência Ambiental (CORNO <i>et al.</i> , 2016)	Politecnico di Torino, na Itália	Estudantes de graduação	Atividades teóricas e práticas	Raspberry Pi, Android, protocolos Web e sistemas <i>smart home</i>	Atividades práticas e exercícios em sala de aula, exercícios guiados no laboratório e trabalhos em grupo no laboratório.

Fonte – o autor.

2.3 Conclusão

Neste capítulo, foram apresentados os principais assuntos, temas e trabalhos relacionados a esta pesquisa. Foi possível explorar o surgimento e a evolução de Computação Ubíqua, assim como seus principais conceitos e características.

Um conjunto de trabalhos mais fortemente relacionados a esta pesquisa foi descrito e discutido. Estes trabalhos trazem soluções que possuem o propósito de auxiliar aulas práticas de Computação Ubíqua. Há também a organização de cursos que propõem uma estrutura curricular para disciplinas relacionadas à Computação Ubíqua.

A proposta de um ambiente para ensino-aprendizagem de Computação Ubíqua neste trabalho tem como objetivo integrar os assuntos apresentados nesta fundamentação teórica para permitir principalmente a prática de conceitos ubíquos estudados em sala de aula.

3 ENSINO DE COMPUTAÇÃO UBÍQUA

Este capítulo apresenta um estudo sobre o ensino e a aprendizagem de Computação Ubíqua no ensino superior de diversas instituições com o intuito de responder à primeira questão desta pesquisa. Essa primeira questão busca conhecer como se dá o ensino de Computação Ubíqua nas universidades, do qual já foi possível ter uma ideia inicial com a apresentação de alguns trabalhos científicos que tratam do tema no Capítulo 2.

O objetivo deste estudo foi verificar a estrutura de alguns cursos de graduação e pós-graduação relacionados ao tema de interesse. Buscou-se averiguar como eles estão organizados e como são executados, como são as suas aulas e quais as maiores dificuldades enfrentadas pelos participantes desses cursos. Nesse contexto, alguns professores e alunos foram contatados para responder a perguntas consideradas relevantes para esta pesquisa.

Com isso, a Seção 3.1 descreve os estudos feitos com professores de diversas instituições de ensino superior. A Seção 3.2 apresenta pesquisas realizadas com estudantes nos níveis de graduação e pós-graduação de várias universidades e faculdades. Por fim, a Seção 3.3 traz as conclusões referentes aos dados obtidos com as respostas de professores e alunos.

3.1 Pesquisa realizada com professores

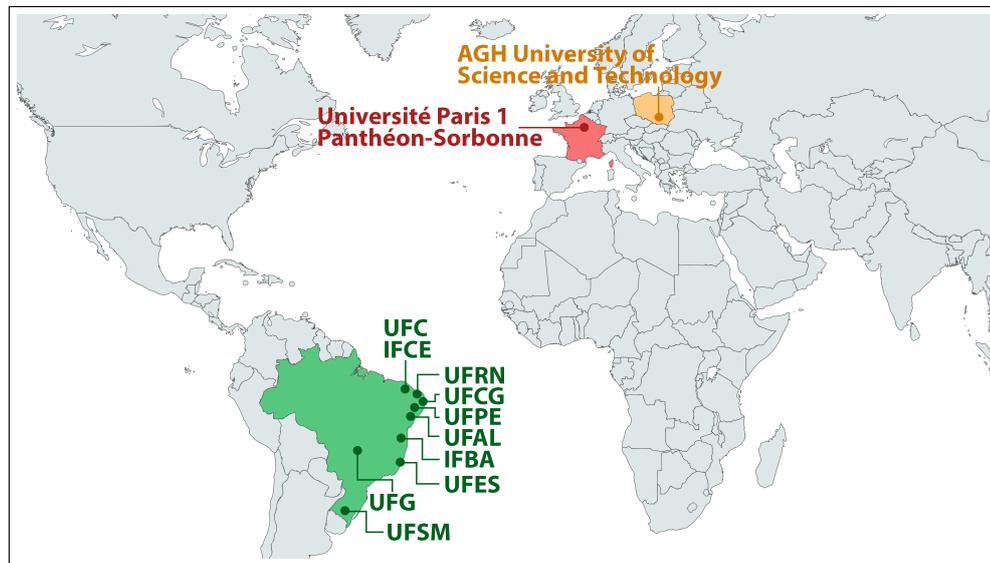
3.1.1 Perfil dos professores

Nesse contexto, foram consultados quinze professores das seguintes universidades: Universidade Federal do Ceará (UFC), Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE), Universidade Federal do Rio Grande do Norte (UFRN), Universidade Federal de Campina Grande (UFCG), Universidade Federal de Pernambuco (UFPE), Universidade Federal de Alagoas (UFAL), Instituto Federal de Educação, Ciência e Tecnologia da Bahia (IFBA), Universidade Federal do Espírito Santo (UFES), Universidade Federal de Goiás (UFG), Universidade Federal de Santa Maria (UFSM), *Université Paris 1 Panthéon-Sorbonne*, na França e *AGH University of Science and Technology*, na Polônia. A Figura 9 ilustra a localização das instituições dos professores que participaram deste estudo.

Esses professores foram escolhidos por conveniência a partir da manifestação de seu interesse em participar deste trabalho quando responderam a mensagens enviadas por e-mail ou à pergunta postada no ResearchGate¹ que indagava sobre qual conceito de Computação Ubíqua

¹ <https://www.researchgate.net/>

Figura 9 – Mapa das universidades dos professores pesquisados.



Fonte – o autor.

seria mais complexo para ensinar. Os e-mails foram conseguidos através da indicação de outros professores ou em *sites* de eventos científicos relacionados ao tema de estudo.

Conforme descreve o Quadro 3, os professores pesquisados apresentam diferentes tempos de experiência no ensino superior. Essa variação de experiência, assim como a diversificação das universidades e das disciplinas ministradas, é interessante para que se possa observar um pouco da similaridade de ideias, organização de trabalho e dificuldades encontradas no ensino de Computação Ubíqua apesar desses fatores diferenciais. O Quadro 3 mostra ainda as disciplinas em que os entrevistados trabalham conteúdos relacionados à Computação Ubíqua.

3.1.2 *Materiais e métodos*

As pesquisas foram feitas de forma presencial em universidades, a distância pelo aplicativo Skype ou ainda por meio de trocas de mensagens via e-mail ou ResearchGate. Uma dessas maneiras de comunicação era escolhida de acordo com a disponibilidade e localização de cada professor. Em todas elas, o professor era guiado para responder às perguntas discursivas de um questionário (Apêndice A) que procurava explorar o cenário das aulas de Computação Ubíqua ministradas. Nas entrevistas realizadas de forma presencial ou por Skype, toda a conversa foi gravada com ciência e autorização do participante, além da tomada de notas por escrito.

O objetivo das entrevistas com os professores foi saber quais de suas disciplinas ensinadas estavam relacionadas à Computação Ubíqua e quais conceitos, na visão deles, eram

Quadro 3 – Perfil dos professores pesquisados.

	Instituição	Experiência no ensino superior	Disciplinas que abordam Computação Ubíqua
Professor 1	UFC	15 anos	Sistemas Distribuídos e Computação Móvel e Ubíqua
Professor 2	UFC	1 ano	Internet das Coisas
Professor 3	UFC	7 anos	Desenvolvimento de <i>Software</i> para Dispositivos Móveis
Professor 4	UFCEG	10 anos	Engenharia de <i>Software</i> para Sistemas Pervasivos
Professor 5	UFES	20 anos	Computação Móvel, Ubíqua e Pervasiva
Professor 6	UFAL	8 anos	Interface Homem-Máquina
Professor 7	<i>Université Paris I Panthéon-Sorbonne</i>	11 anos	Informática Sensível ao Contexto, Cooperação e Ubiquidade e Informática Pervasiva e <i>Cloud Computing</i>
Professor 8	IFBA	15 anos	Aspectos Avançados em Desenvolvimento de Sistemas e Engenharia de Sistemas Ubíquos
Professor 9	IFCE	4 anos	Sistemas Distribuídos
Professor 10	UFPE	6 anos	Sistemas de Informação Urbanos e Tópicos Avançados em Sistemas de Informação
Professor 11	UFPE	22 anos	Sistemas Distribuídos
Professor 12	UFRN	16 anos	Sistemas Distribuídos, Tópicos Especiais em Redes e Computação Móvel
Professor 13	UFES	16 anos	Sistemas e Aplicações Multimídia
Professor 14	UFG	11 anos	Engenharia de <i>Software</i> Aplicada à Web Semântica
Professor 15	<i>AGH University of Science and Technology</i>	7 anos	<i>Ambient Intelligence</i>

Fonte – o autor.

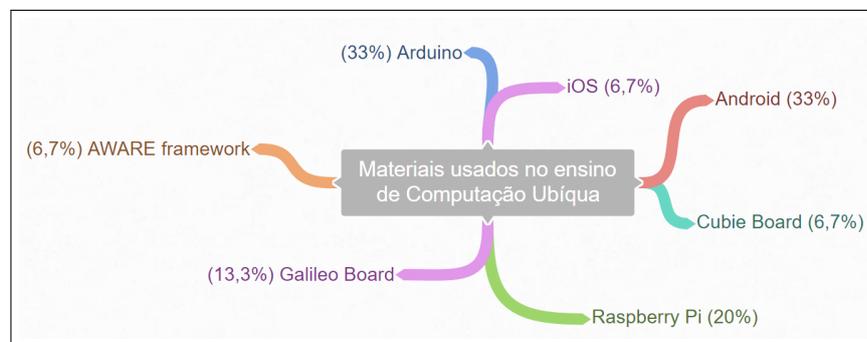
considerados mais importantes. Também procurou-se identificar a experiência do professor, a organização do curso, os materiais usados em aulas práticas, o número de atividades práticas, as dificuldades relacionadas ao ensino de conceitos de Computação Ubíqua e como eles imaginavam um ambiente para a prática de Computação Ubíqua. Para relatar este estudo qualitativo, usou-se como base de apoio os itens da lista de verificação COREQ (TONG *et al.*, 2007), que tem como proposta auxiliar na descrição de estudos qualitativos por meio de 32 itens específicos.

3.1.3 Resultados

Dentre as questões apresentadas no questionário do Apêndice A, apenas as de maior relevância para esta discussão estão discriminadas a seguir. As imagens abaixo contêm as respostas mais recorrentes dos professores, procurou-se evitar repetições.

Dos 15 professores consultados, 46% ministram suas disciplinas tanto na graduação como na pós-graduação, 27% somente na graduação e 27% apenas na pós-graduação. 73% dos professores dividem suas disciplinas em aulas teóricas e práticas e 27% deles não possuem aulas práticas em sala de aula ou laboratório, mas fazem trabalhos de desenvolvimento de aplicações relacionadas ao tema estudado como atividades extraclasse. 47% dos professores utilizam algum recurso para auxiliá-los no desenvolvimento de suas aulas práticas. A Figura 10 mostra alguns desses recursos utilizados, sendo Arduino e Android os mais recorrentes, aparecendo nas aulas práticas de 33% dos professores. Quando perguntados sobre os problemas encontrados no ensino de Computação Ubíqua, os professores falaram principalmente das dificuldades em adquirir materiais para a realização das aulas práticas – ou por falta de verba da instituição ou pela inexistência de ferramenta para um propósito específico –, da complexidade de alguns conceitos ubíquos, da complicação em aplicar esses conceitos à prática e da má qualidade na formação básica dos alunos, o que às vezes demanda mais aulas para explicação de determinados tópicos, impedindo o avanço planejado do curso. A Figura 11 apresenta as principais dificuldades enfrentadas por esses professores.

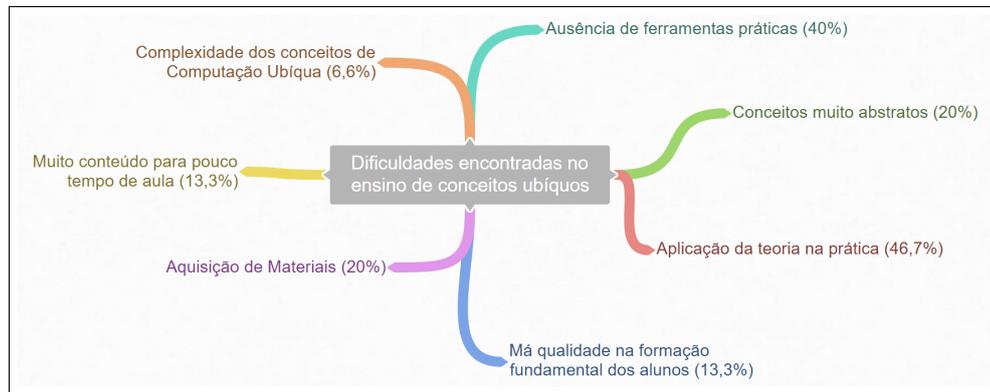
Figura 10 – Materiais usados em aulas práticas.



Fonte – o autor.

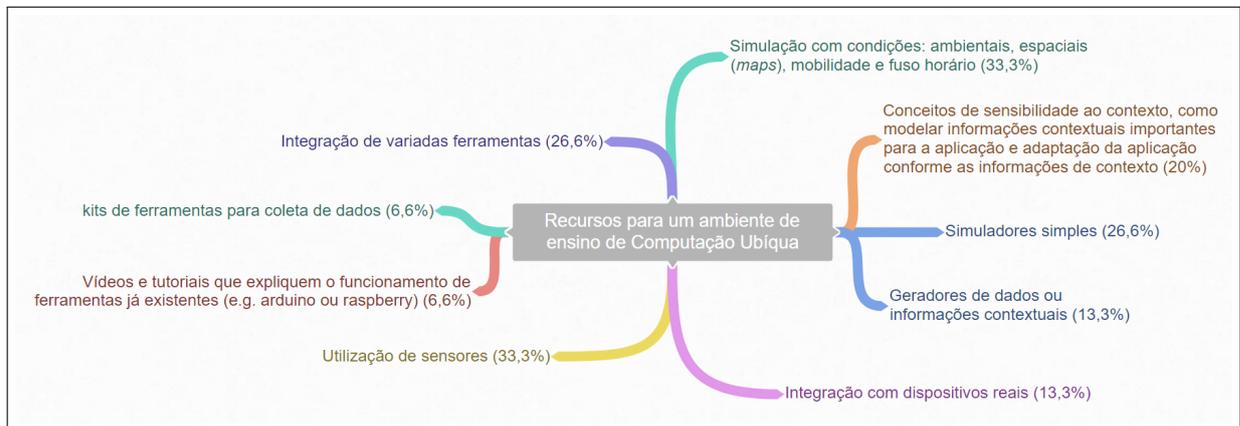
Quando indagados sobre o que deveria ter em um ambiente voltado para o ensino de conceitos de Computação Ubíqua, os professores deram diversas respostas que estão apresentadas na Figura 12. Com relação ao conceito ubíquo indispensável a esse ambiente, 80% respondeu que seria Sensibilidade ao Contexto. A Figura 13 apresenta os conceitos considerados mais importantes pelos professores.

Figura 11 – Dificuldades encontradas no ensino de Computação Ubíqua.



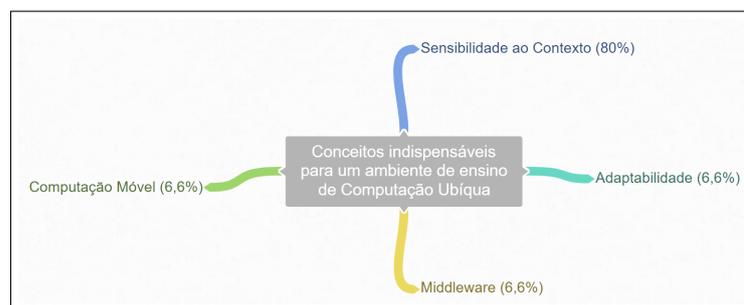
Fonte – o autor.

Figura 12 – Recursos que deveriam existir em um ambiente de ensino para Computação Ubíqua.



Fonte – o autor.

Figura 13 – Conceitos indispensáveis para um ambiente de ensino para Computação Ubíqua.



Fonte – o autor.

3.1.4 Análise e discussões

Com a pesquisa realizada com os professores, foi possível confirmar a necessidade de uma ferramenta ou ambiente para o ensino-aprendizagem de Computação Ubíqua. Essa necessidade não foi identificada como algo local, mas que se estende a várias outras instituições

de ensino superior presentes no Brasil e fora dele. Os professores falaram ainda que há conceitos abstratos e complexos o bastante para dificultar ou comprometer o desenvolvimento de atividades práticas.

Alguns requisitos funcionais de usuário foram então coletados conforme as respostas dos professores. O Quadro 4 apresenta esses requisitos, que foram utilizados para a estruturação do ambiente Lucy. Apenas o requisito RF03 não foi totalmente satisfeito na versão do ambiente entregue por esta dissertação em virtude das limitações de tempo.

Quadro 4 – Requisitos funcionais de usuário definidos pelos professores.

Requisitos funcionais de usuário
RF01. O ambiente deve ter simulações que exemplifiquem os conceitos abordados.
RF02. O ambiente deve ilustrar a utilização de informações contextuais em aplicações.
RF03. O ambiente deve possuir a integração de variadas ferramentas, assim como possibilitar seu uso dentro dele.
RF04. O ambiente deve possuir áreas para conteúdos teóricos e trabalhos relacionados aos conceitos estudados.
RF05. O ambiente deve proporcionar a utilização de sensores.

Fonte – o autor.

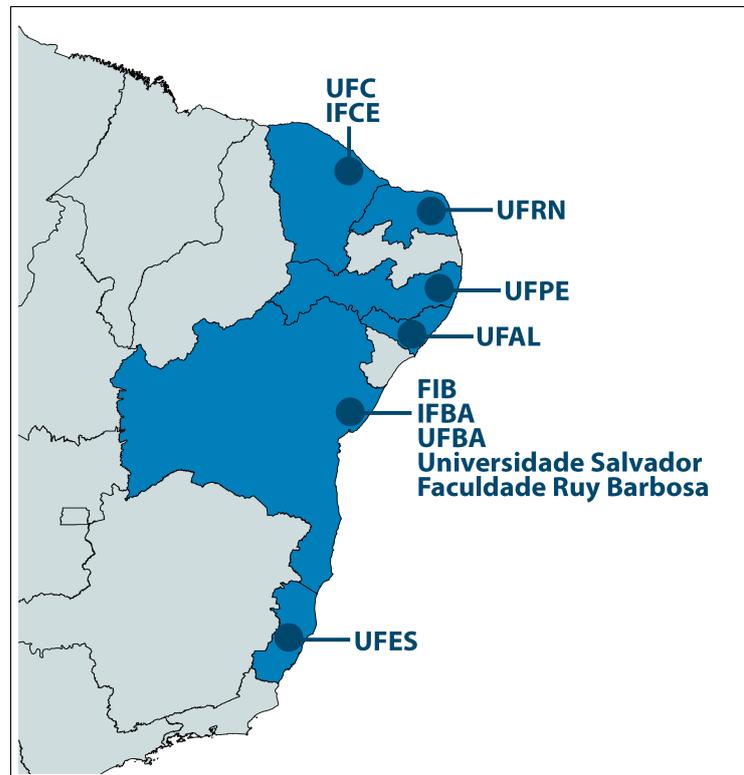
3.2 Pesquisa realizada com estudantes

3.2.1 Perfil dos estudantes

Obteve-se então o total de 60 respostas de estudantes da graduação e da pós-graduação, sendo 32 de alunos da UFC e 28 de alunos de outras instituições de ensino superior, tais como IFCE, UFRN, UFPE, UFAL, Faculdade Integrada da Bahia (FIB), IFBA, Universidade Federal da Bahia (UFBA), Universidade Salvador, Faculdade Ruy Barbosa e UFES. A Figura 14 ilustra a disposição das universidades e faculdades dos participantes desta pesquisa.

Todos os estudantes participantes eram de cursos ligados à tecnologia. Uma análise mais detalhada das respostas desses alunos será feita a seguir. Pelo fato do número de respostas da UFC ser maior, separou-se as informações obtidas em dois grupos, alunos da UFC e de outras instituições. Dentro de cada grupo citado, diferenciou-se também as respostas obtidas de estudantes da graduação e da pós-graduação. Dos 32 estudantes da UFC que responderam ao questionário, 58.1% eram da pós-graduação e 41.9% da graduação. Dos 28 alunos consultados nas diferentes instituições de ensino superior, 48.3% eram da pós-graduação e 51.7% da graduação.

Figura 14 – Mapa de localização das instituições de ensino dos estudantes pesquisados.



Fonte – o autor.

3.2.2 *Materiais e métodos*

Para as pesquisas com estudantes, foi aplicado um questionário (Apêndice B) composto de questões, em sua maioria, objetivas, caracterizando este estudo mais como quantitativo. Em algumas dessas questões, podia-se marcar mais de um item como resposta, o que gerou resultados cuja soma é superior a 100% para cada uma dessas questões. Esse questionário foi aplicado de forma *online*, via grupos de disciplinas no Facebook que tinham alguma relação com Computação Ubíqua ou que eram exclusivamente sobre o assunto. Além disso, foi pedido também para os professores entrevistados compartilhar o questionário em listas e grupos de cursos relacionados à Computação Ubíqua.

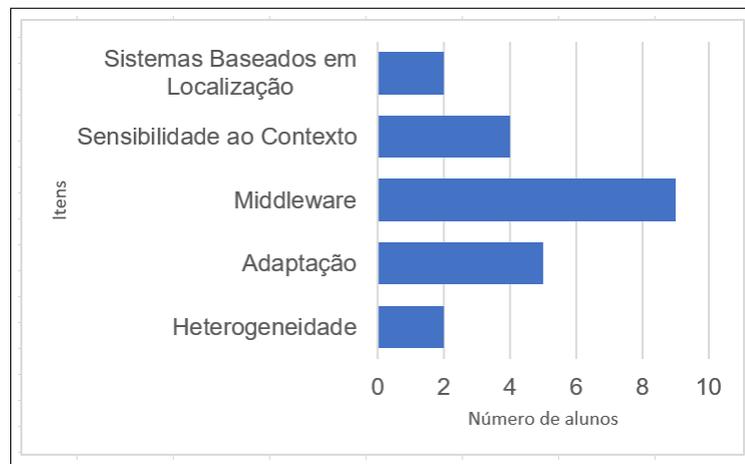
O objetivo do questionário foi saber dos alunos se eles possuíam algum nível de experiência em programação no momento da realização da disciplina, se tinham aulas práticas de Computação Ubíqua, se usavam alguma ferramenta para essas aulas práticas, quais os conceitos que eles mais tinham dificuldade de entender e o que eles esperavam de um ambiente proposto para o ensino de Computação Ubíqua. No final do questionário, havia um espaço para sugestões ou observações a respeito de aulas práticas no ensino de Computação Ubíqua.

3.2.3 Resultados dos alunos da UFC

Do total de alunos da UFC pesquisados, 80.6% tiveram aulas práticas de Computação Ubíqua, as quais foram realizadas em sua maioria com a utilização de Android.

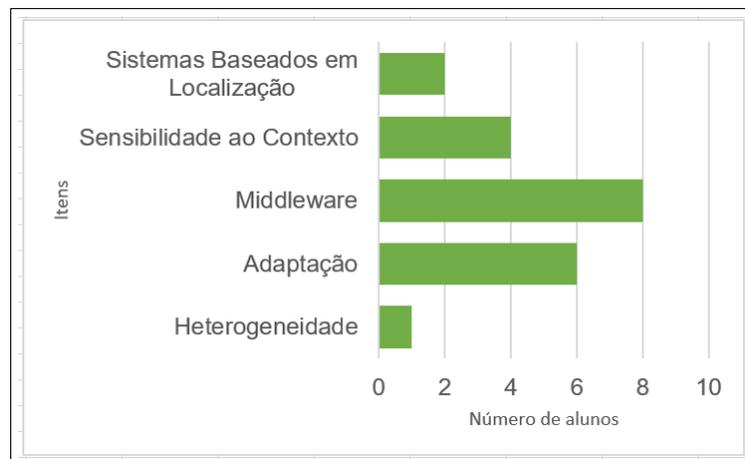
Quando perguntados sobre quais conceitos ubíquos eram mais difíceis de se entender na ausência de aulas práticas, tanto os alunos da pós-graduação como os da graduação responderam que “*Middleware*” era o mais complicado, conforme ilustram as Figuras 15 e 16 respectivamente.

Figura 15 – Conceitos considerados difíceis para alunos de pós-graduação da UFC.



Fonte – o autor.

Figura 16 – Conceitos considerados difíceis para alunos de graduação da UFC.

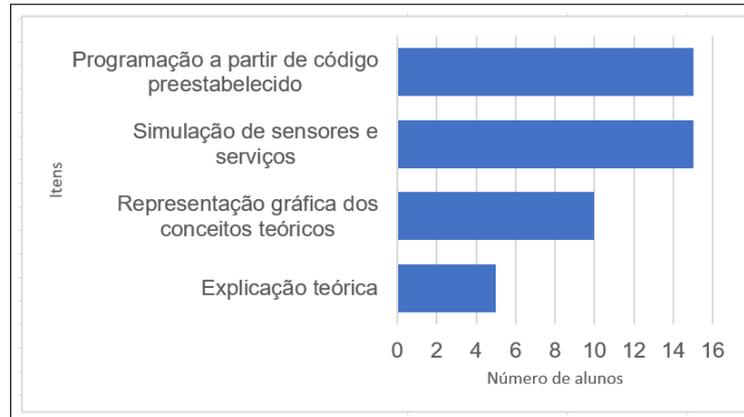


Fonte – o autor.

Quando indagados sobre o que deveria ter em um ambiente para o ensino de conceitos ubíquos, os estudantes da pós-graduação e da graduação escolheram preferencialmente

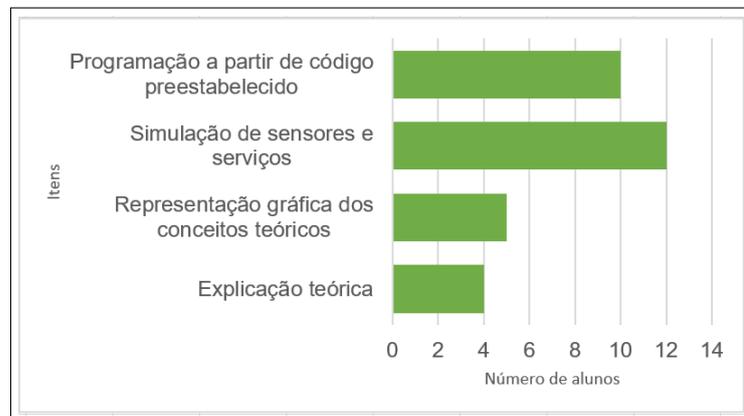
“Programação a partir de código preestabelecido” e “Simulação de sensores e serviços” como apresentam as Figuras 17 e 18.

Figura 17 – Recursos necessários para um ambiente de ensino de Computação Ubíqua segundo alunos de pós-graduação da UFC.



Fonte – o autor.

Figura 18 – Recursos necessários para um ambiente de ensino de Computação Ubíqua segundo alunos de graduação da UFC.



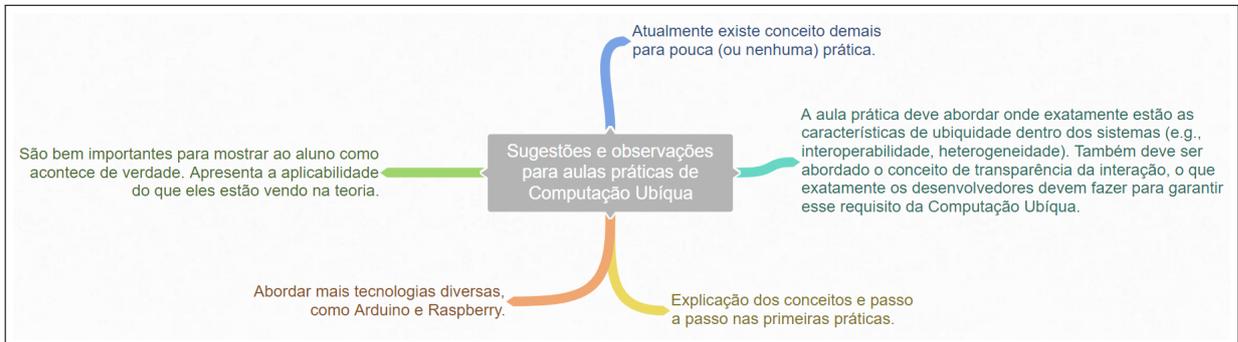
Fonte – o autor.

Alguns estudantes escreveram ainda sugestões ou observações sobre aulas práticas no ensino de Computação Ubíqua. Essas informações estão apresentadas na Figura 19.

3.2.4 Resultados dos alunos de diferentes instituições de ensino superior

Do total de estudantes de outras instituições pesquisados, 37.9% tiveram aulas práticas de Computação Ubíqua, as quais foram realizadas com o auxílio de diversos materiais

Figura 19 – Sugestões e observações de alunos da UFC sobre aulas práticas de Computação Ubíqua.

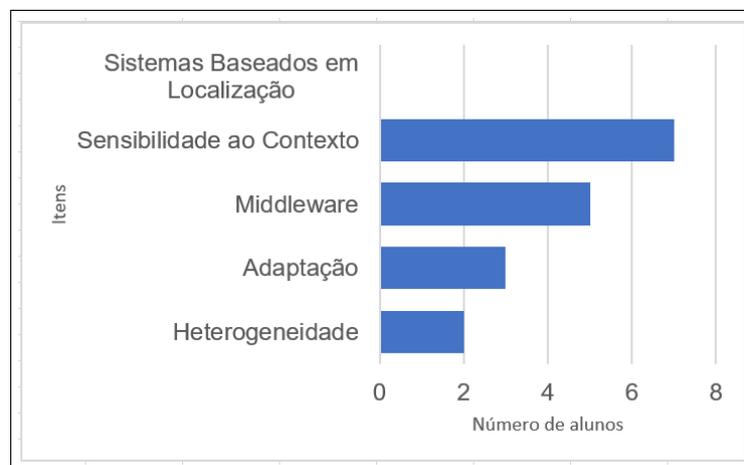


Fonte – o autor.

como Arduino, Raspberry Pi, Android e Galileo Board.

Quando perguntados sobre quais conceitos ubíquos eram mais difíceis de se entender na ausência de aulas práticas, os alunos da pós-graduação responderam que era “Sensibilidade ao Contexto” e os da graduação responderam que “*Middleware*” era o mais complicado, conforme ilustram as Figuras 20 e 21 respectivamente.

Figura 20 – Conceitos considerados difíceis para alunos de pós-graduação de outras instituições.

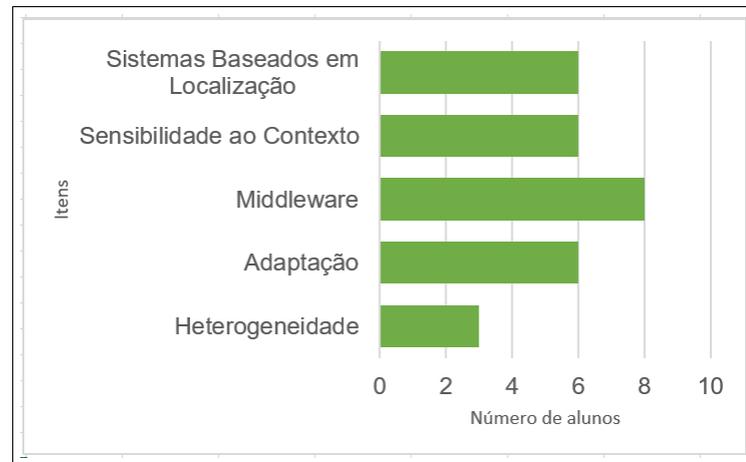


Fonte – o autor.

Quando indagados sobre o que deveria ter em um ambiente para o ensino de conceitos ubíquos, os estudantes da pós-graduação e da graduação dessas outras instituições, assim como os da UFC, escolheram preferencialmente “Programação a partir de código preestabelecido” e “Simulação de sensores e serviços” como apresentam as Figuras 22 e 23.

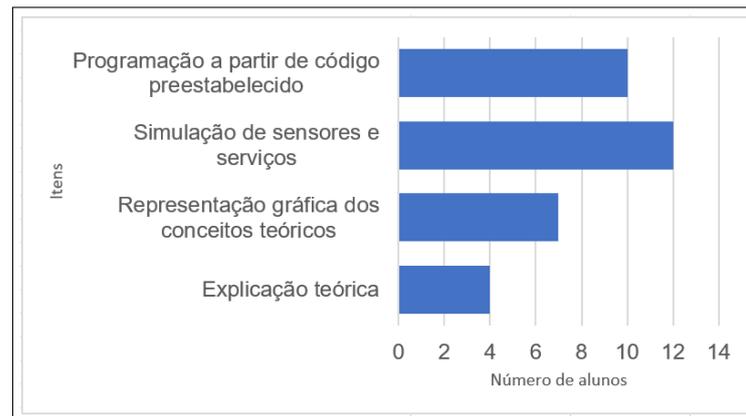
Alguns estudantes dessas outras instituições escreveram também sugestões ou observações sobre aulas práticas no ensino de Computação Ubíqua. Essas informações estão

Figura 21 – Conceitos considerados difíceis para alunos de graduação de outras instituições.



Fonte – o autor.

Figura 22 – Recursos necessários para um ambiente de ensino de Computação Ubíqua segundo alunos de pós-graduação de outras instituições.



Fonte – o autor.

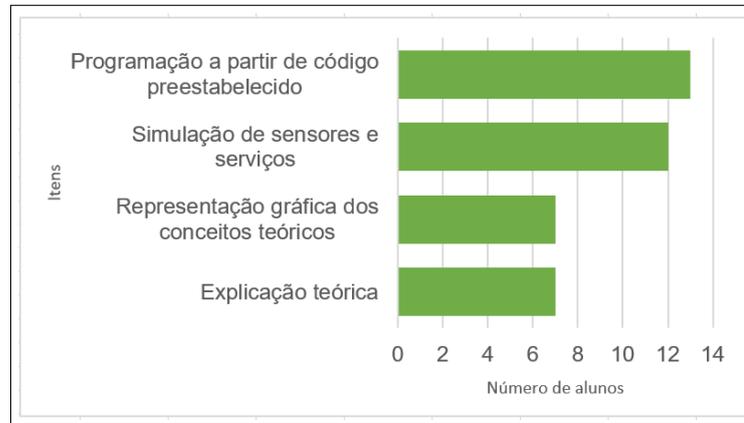
apresentadas na Figura 24.

3.2.5 *Análise e discussões*

Segundo os estudantes, as aulas práticas, necessariamente, precisam estar alinhadas com a teoria dada a fim de garantir a aplicação do conhecimento adquirido, o que, segundo eles, nem sempre acontecia em suas disciplinas. Observações dessa natureza foram feitas na seção do questionário destinada a sugestões e observações, mas a maioria dos alunos optou por não preencher esse espaço.

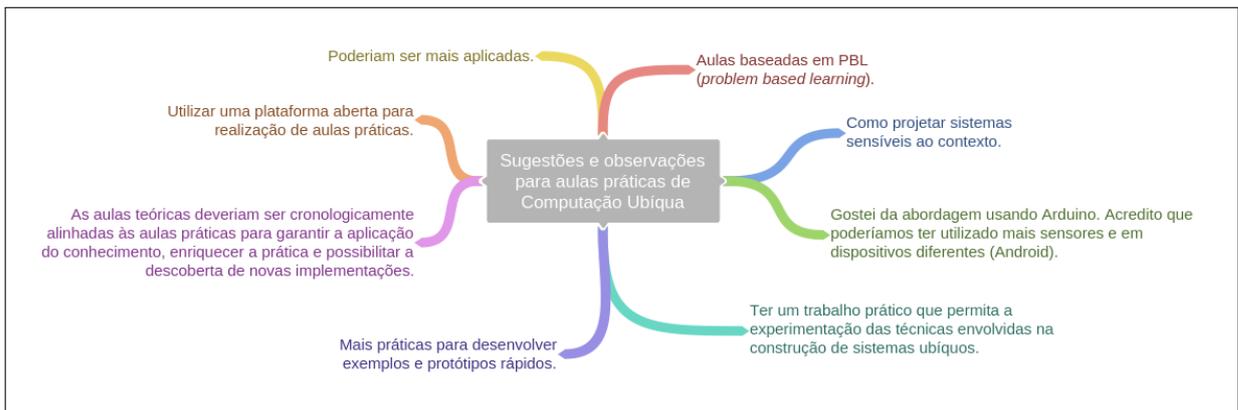
O Quadro 5 apresenta alguns requisitos funcionais destacados pelos estudantes. Esses requisitos foram utilizados como base para a estruturação do ambiente Lucy.

Figura 23 – Recursos necessários para um ambiente de ensino de Computação Ubíqua segundo alunos de graduação de outras instituições.



Fonte – o autor.

Figura 24 – Sugestões e observações de alunos de outras instituições de ensino superior sobre aulas práticas de Computação Ubíqua.



Fonte – o autor.

Quadro 5 – Requisitos funcionais de usuário definidos pelos alunos.

Requisitos funcionais de usuário
RF01. O ambiente deve ter programação a partir de código preestabelecido.
RF02. O ambiente deve ter simulação de sensores e serviços.
RF03. O ambiente deve possuir representação gráfica dos conceitos teóricos.
RF04. O ambiente deve ter explicação teórica.

Fonte – o autor.

3.3 Conclusão

Este capítulo apresentou uma pesquisa sobre o cenário de cursos relacionados à Computação Ubíqua em algumas instituições de ensino superior no Brasil e fora dele. A divisão das disciplinas entre aulas teóricas e práticas é algo que acontece com bastante frequência e a

porcentagem do curso dedicada a uma dessas modalidades varia conforme seus objetivos.

Existem diferentes ferramentas sendo usadas nas aulas práticas, mas tanto professores quanto alunos relataram a falta de artefatos mais específicos voltados para a prática de determinados conceitos de Computação Ubíqua. O público pesquisado relatou alguns desses conceitos que devem ser abordados pelo Lucy, destacando como principais Sensibilidade ao Contexto e *Middleware*.

Para professores e estudantes, a ideia de um ambiente que possa auxiliar na prática de conceitos de Computação Ubíqua foi muito bem recebida. Com isso, eles relataram alguns recursos que seriam necessários a um ambiente dessa natureza, tais como integração com dispositivos reais, simulações com condições determinadas por meio da inserção de parâmetros, explicação teórica de conceitos, exemplos de códigos para a programação e construção de aplicações rápidas relacionadas ao tema estudado e integração de diferentes ferramentas. Além disso, é preciso que sempre se estabeleça a relação entre o conteúdo teórico estudado e o que é aplicado na prática.

4 LUCY - LEARNING UBIQUITOUS COMPUTING EASILY

Este capítulo apresenta o Lucy, um ambiente desenvolvido para auxiliar as aulas práticas de Computação Ubíqua. Com base no que é descrito nos Capítulos 2 e 3, é possível responder à segunda questão desta pesquisa, que pergunta das características necessárias a um ambiente que favoreça a compreensão dos estudantes acerca de conceitos relacionados á Computação Ubíqua. Professores e alunos mencionaram, no Capítulo 3, diversos recursos para compor esse ambiente. Nesse sentido, o ambiente criado oferece componentes que possibilitam revisão e consulta de materiais teóricos, experimentação e simulação do funcionamento de conceitos ubíquos estudados e programação e correção de códigos Android para a construção de pequenas aplicações. O Lucy se utiliza de *smartphones* e seus sensores para prover observação do comportamento de conceitos ubíquos conforme variação dos dados obtidos pelos sensores do próprio *smartphone*.

A Seção 4.1 mostra um cenário motivador para ilustrar as possibilidades de abordagem e de utilização do Lucy no ensino de conceitos ubíquos. A Seção 4.2 traz a descrição de implementação e de organização da estrutura do Lucy. Por fim, a Seção 4.3 expõe as conclusões a respeito da proposta desenvolvida e apresentada neste capítulo.

4.1 Cenário motivador

A fim de ilustrar o potencial da abordagem de conceitos ubíquos e de utilização do Lucy, um cenário motivador foi criado e apresentado a seguir. Os professores usam os mais diversos materiais para suas aulas práticas e as maneiras de utilização varia conforme os objetivos do professor e da disciplina. O cenário motivador a seguir mostra uma das formas de uso do Lucy.

4.1.1 Uma aula prática de *Middleware*

O professor Lesley possui uma disciplina de Computação Ubíqua na graduação de uma universidade pública. Lesley começa uma aula teórica sobre *Middleware*, explica os conceitos de *Middleware*, detalha as diferenças entre um *Middleware* tradicional e um *Middleware* para Computação Ubíqua e dá alguns exemplos daquilo que fala referenciando OSBI e Java RMI. Lesley explica, por exemplo, os conceitos de desacoplamento temporal e transparência de comunicação relevantes para sistemas voláteis, como os ubíquos.

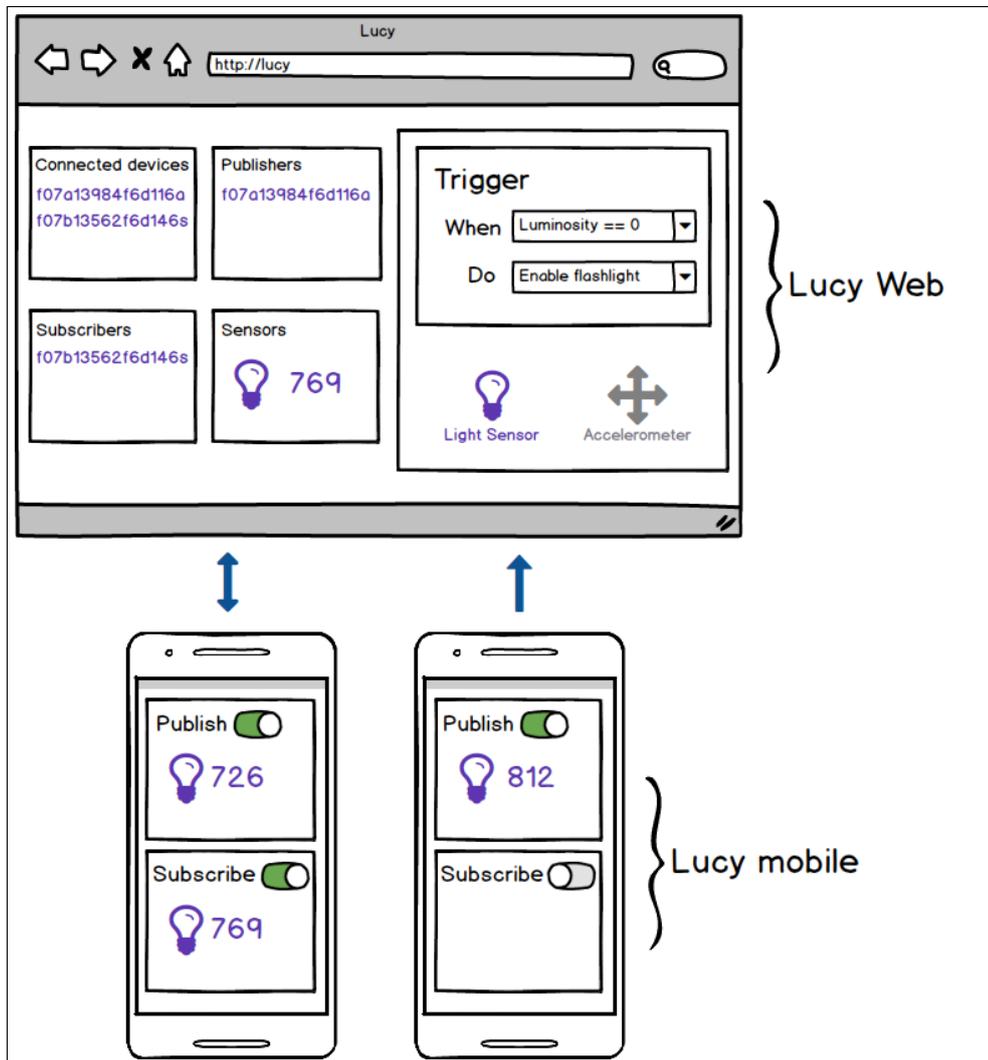
Lesley quer poder mostrar para seus alunos o funcionamento prático daquilo que ele falou durante a aula teórica. Ele quer que seus alunos experimentem e simulem alguns comportamentos explicados em sala e que consigam tirar as conclusões necessárias para compreender o que é e como funciona um *Middleware*. Lesley combina então com seus alunos que a aula seguinte será para praticar, com o auxílio do Lucy, as características e comportamentos do conceito discutido naquele dia.

Na aula prática, o professor Lesley abre a seção de simulação do Lucy e então pede para que seus alunos acessem o Lucy móvel instalado nos *smartphones* deles e o conectem à chave de acesso disponibilizada pela simulação aberta pelo professor. Feito isso, Lesley explica que o objetivo da simulação é ilustrar o funcionamento de um *Middleware* por meio do modelo de comunicação indireta *publish/subscribe* abordado pelo Lucy. A simulação deve então ajudar os alunos a experimentarem as possibilidades desse tipo de *Middleware*.

Conforme apresenta a Figura 25, o Lucy Web mostra, do lado direito, os sensores que podem ser habilitados e desabilitados, além das regras que podem ser criadas para cada um deles. Do lado esquerdo, há informações dos dispositivos que estão conectados, dos que estão inscritos para receber informações e dos que estão publicando informações. Além disso, há um quadro que mostra um valor relativo aos sensores habilitados, esse valor é a média dos valores fornecido pelos sensores dos celulares.

Lesley habilita o sensor de luminosidade na aplicação Web, insere uma regra para que, quando a luminosidade for igual a 0, a lanterna dos *smartphones* conectados seja ligada. Então, pede para que todos os alunos ativem a opção "*publish*" de suas aplicações e apenas alguns habilitem o "*subscribe*". Lesley mostra que todos estão fornecendo valores de luminosidade diferentes provenientes do sensor de luz de seus dispositivos e que a aplicação Web calcula a média dos valores recebidos, apresentando o resultado em sua tela no quadro de "*sensors*". As aplicações que estão inscritas para receber as informações da página Web recebem o cálculo da média como novo valor de luminosidade a ser considerado pelo celular. Assim, Lesley pede para que seus alunos cubram o sensor de luminosidade do seu aparelho até o valor correspondente a ele ser zerado. Quando o valor apresentado na tela do Lucy Web chega a 0, a lanterna dos celulares da turma é ativada. Para os dispositivos que não estão com a opção de "*subscribe*" ativa, Lesley pede para que apenas seus possuidores cubram o sensor de luminosidade até este chegar em 0. Os alunos percebem então que, mesmo que o valor apresentado na tela do Lucy Web não seja 0, a lanterna dos celulares com o "*subscribe*" desabilitado foi acesa. Isso acontece

Figura 25 – Esquema do funcionamento da simulação de *Middleware* do Lucy.



Fonte – o autor.

porque, como esses celulares não estão recebendo valores de luminosidade provenientes do Lucy Web, eles consideram unicamente os valores recebidos pelos seus próprios sensores para disparar a ação ditada pela regra criada pelo professor no Lucy Web.

Depois dessa simulação, Lesley pede para que cada aluno se cadastre no ambiente e configure novas regras dentro da simulação. Para que os estudantes consigam verificar o funcionamento das regras com a utilização de dispositivos diferentes, o professor pede para que eles formem duplas ou trios. Caso alguém tenha esquecido algo relativo à aula teórica de *Middleware*, há artigos, *slides*, vídeos e infográficos na seção de *Lecture* que podem ajudar a sanar as dúvidas dos alunos. Como exercício para casa, os estudantes devem acessar a seção de *Practice*, baixar o código-fonte de uma aplicação criada a partir do padrão *publish/subscribe*, corrigir os erros que ela apresenta e acrescentar novos sensores e regras. Com essa simulação,

Lesley ilustra os conceitos de publicação e subscrição de eventos (*Publish/Subscribe*) numa interação entre componentes. Esses conceitos são importantes para a Computação Ubíqua porque promovem o desacoplamento referencial em sistemas distribuídos, o que permite o estabelecimento de um canal de comunicação sem o conhecimento do endereço de rede das outras entidades.

Conforme a definição de simulação na Seção 2.2 do Capítulo 2, o que foi descrito mais acima, referente a Figura 25, enquadra-se como simulação por se caracterizar como a representação do comportamento e do funcionamento de um *Middleware*, não sendo um *Middleware* de fato.

4.2 Estrutura do Lucy

Há uma gama de ambientes voltados para o ensino e a aprendizagem dos mais diversos temas e conteúdos, tais como Solar¹, Moodle², Sakai³ e Teleduc⁴. Existem os que se utilizam de determinadas áreas da Computação a fim de apoiar atividades de outras áreas, como o trabalho proposto por (SARMENTO *et al.*, 2012), o qual trata de um laboratório que usa Computação Ubíqua para apoiar atividades laboratoriais relacionadas à Física e à Química, por exemplo.

Existem também ambientes dedicados especialmente a tópicos de Computação e áreas afins. Com isso, há ambientes mais específicos, como (OLIVEIRA; SANTOS, 2016) e (CACHO *et al.*, 2016), com materiais e abordagens direcionados a temas como Redes, Lógica de Programação, Internet das Coisas ou Computação Ubíqua, por exemplo.

Ambiente, conforme explica (SANTOS, 2003), pode ser entendido como tudo aquilo que engloba pessoas, natureza ou coisas e objetos técnicos. Ambientes de aprendizagem, então, podem ser classificados como espaços nos quais ocorre um processo sócio-técnico em que os sujeitos interagem para a construção de saberes e conhecimentos (SANTOS, 2003).

Lucy é um ambiente, segundo as definições de (SANTOS, 2003), Web com acesso a *smartphone* e que tem o objetivo de apoiar professores e alunos em aulas práticas de Computação

¹ <http://www.solar.virtual.ufc.br/>

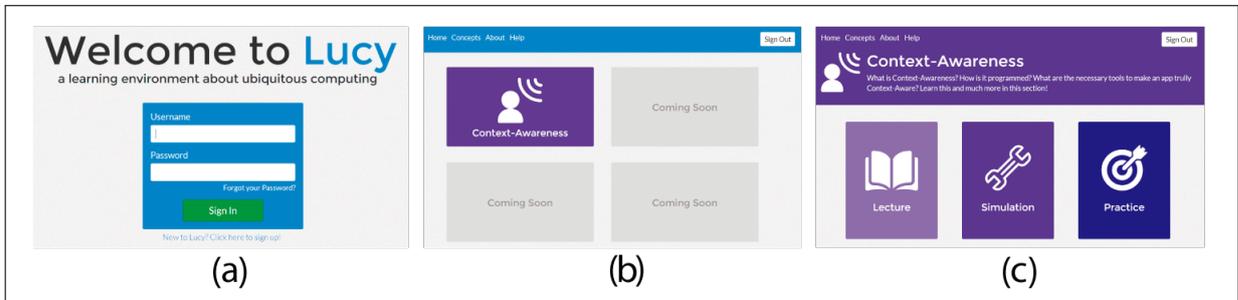
² <https://moodle.org/>

³ www.sakaiproject.org

⁴ www.teleduc.org.br

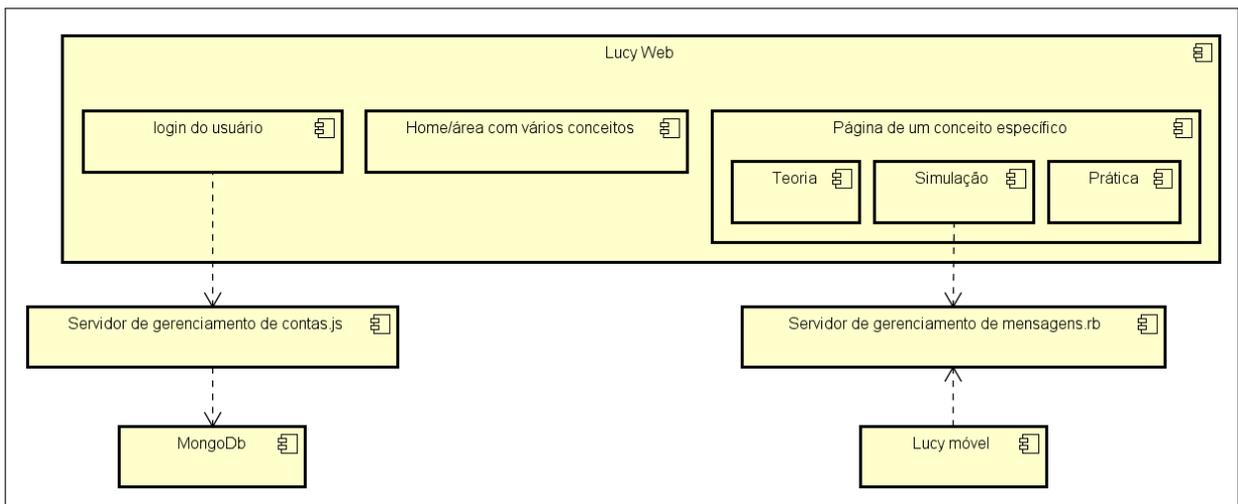
Ubíqua. Ele foi desenvolvido com HTML⁵, CSS⁶, Node.js⁷ e utiliza o MongoDB⁸ como programa de banco de dados. Para acessar o sistema, cada participante deverá criar um usuário e uma senha conforme solicita a tela de acesso mostrada na Figura 26 (a). Depois de acessar o ambiente, o usuário irá encontrar os conceitos relacionados à Computação Ubíqua disponíveis, mostrados na Figura 26 (b). Ao acessar um dos conceitos disponíveis, o estudante ou professor será levado para uma área dividida em três seções: *Lecture*, *Simulation* e *Practice*, como apresenta a Figura 26 (c). A Figura 27 exhibe os principais componentes que integram o Lucy.

Figura 26 – (a) Tela de acesso do Lucy. (b) Tela de exibição dos conceitos ubíquos disponíveis. (c) Área interna de um conceito com divisão em *Lecture*, *Simulation* e *Practice*.



Fonte – o autor.

Figura 27 – Diagrama de componentes do Lucy.



Fonte – o autor.

Como o objetivo do ambiente é apoiar professores e alunos na prática de diversos

⁵ <https://www.w3.org/html/>

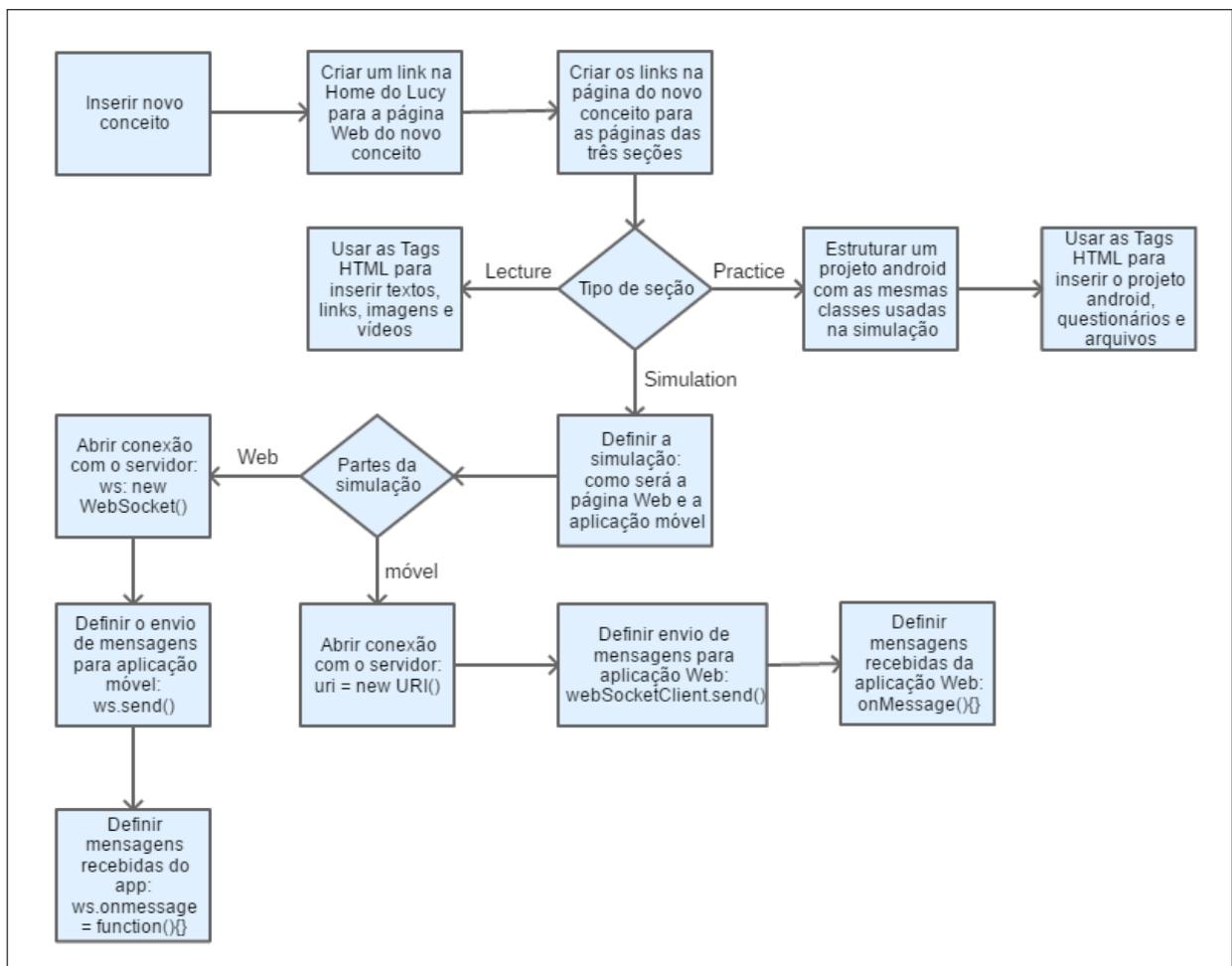
⁶ <https://www.w3.org/Style/CSS/Overview.en.html>

⁷ <https://nodejs.org/en/>

⁸ <https://www.mongodb.com/>

conceitos ubíquos, serão ainda criados e acrescentados a esse sistema outros conceitos considerados importantes por professores e alunos no Capítulo 3, tais como *Middleware*, Adaptabilidade, Heterogeneidade, Computação Móvel e outros. No entanto, para o fechamento do escopo desta pesquisa, foram criados apenas os materiais e as seções relativos à Sensibilidade ao Contexto, pois este foi um dos conceitos mencionado por professores e alunos como essencial ao estudo de Computação Ubíqua. A Figura 28 descreve os passos necessários para a inserção de um novo conceito relacionado à Computação Ubíqua no Lucy.

Figura 28 – Fluxograma para a inserção de um novo conceito no Lucy.



Fonte – o autor.

4.2.1 Lecture

Lecture é uma das seções que compõe cada conceito presente no ambiente Lucy. Nesta seção, o aluno pode encontrar materiais teóricos relacionados a cada tema estudado. Assim, pode-se achar artigos científicos, notas de aulas, *slides*, vídeos, infográficos e outros materiais

teóricos referentes aos conceitos ubíquos.

A alocação desses materiais dentro do ambiente ainda precisa ser feita por alguém que conheça minimamente HTML, pois será necessário acessar o código-fonte da página para disponibilizar os *hiperlinks* pretendidos. Uma tela de edição desses arquivos deverá ser criada e disponibilizada para professores futuramente.

4.2.2 *Simulation*

A seção de *Simulation*, presente dentro de cada conceito, foi criada para permitir que professores e alunos experimentem o comportamento e o funcionamento de cada conceito ubíquo a partir de dados fornecidos pelos sensores de *smartphones*.

A proposta de simulação envolve duas partes: uma Web e uma móvel. A primeira é a simulação do Lucy propriamente dita, a página Web em que há a configuração de parâmetros. A móvel é uma aplicação Android em que se observa o resultado da configuração de parâmetros feita. Para a comunicação entre esses dois componentes, existe um servidor WebSocket em Ruby que gerencia a troca de mensagens entre eles. O WebSocket foi escolhido porque sempre fornece uma conexão persistente e de baixa latência entre o cliente e o servidor, conforme especifica (MELNIKOV; FETTE, 2011). Assim, o endereço IP e a porta onde está sendo executado o servidor são colocados no Lucy móvel e na página Web da simulação do Lucy a fim de estabelecer a comunicação entre essas duas partes. Então, para abrir a conexão do lado Web da simulação, é necessário a utilização do Código-fonte 1. Para estabelecer a conexão do Lucy móvel com o servidor, é necessário o Código-fonte 2 na aplicação Android.

Código-fonte 1 – Abertura da conexão com o servidor na página Web de simulação do Lucy.

```
1 ws = new WebSocket("ws://" + location.hostname + ":8080/");
```

Código-fonte 2 – Abertura da conexão com o servidor na aplicação Android do Lucy.

```
1 uri = new URI("ws://websockethost:8080");
2 websocketClient = new WebSocketClient(uri){ }
```

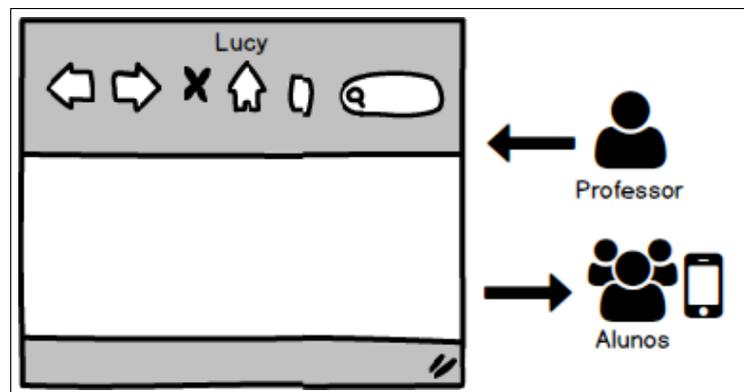
Estabelecida a comunicação entre a simulação do Lucy e o Lucy móvel, é necessária a implementação das funções para checar a conexão aberta ("*onOpen*"), gerenciar as mensagens

recebidas (“*onMessage*”) e encerrar a conexão (“*onClose*”). Para o envio de mensagens, o método (“*send()*”) deve ser implementado. Com essas especificações definidas, é possível se criar uma estrutura Web que funcione como simulação para um conceito ubíquo e uma aplicação Android que receba as mensagens de configuração da simulação Web e apresente o resultado obtido na tela do dispositivo móvel para o usuário.

4.2.2.1 Situações de utilização do simulador do Lucy

A utilização da simulação do Lucy pode ocorrer de três maneiras conforme os objetivos do professor e da atividade a ser executada. Na Figura 29, é apresentado um cenário em que o professor abre a simulação em seu computador e todos os alunos se conectam a esta simulação por meio de uma nova chave de acesso. Em virtude disso, todas as configurações que forem feitas pelo professor em seu simulador serão refletidas nos dispositivos conectados de todos os alunos.

Figura 29 – Situação de uso do simulador pelo professor para vários alunos.

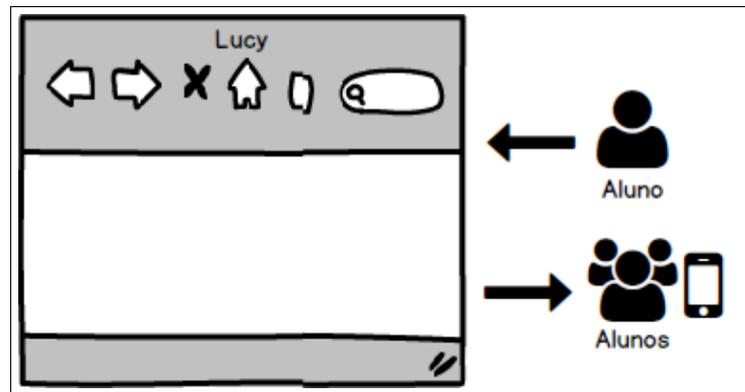


Fonte – o autor.

A situação mostrada pela Figura 30 ilustra uma proposta de atividade em grupo. Então o simulador é aberto na máquina de apenas um dos estudantes e todos os integrantes do grupo conectam seu celular, por meio da chave de acesso, ao simulador aberto.

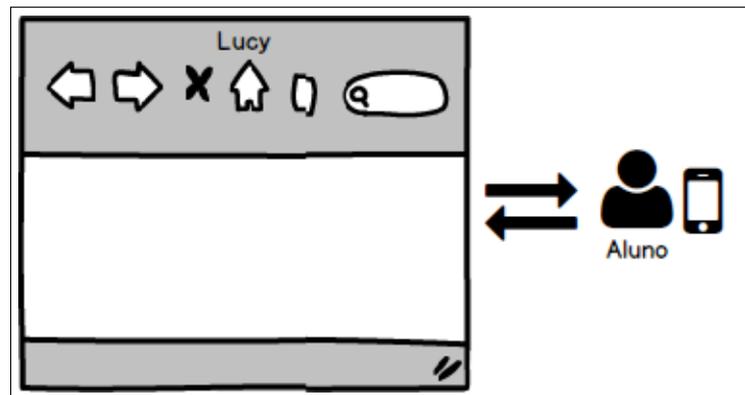
No cenário trazido pela Figura 31, a situação expõe a utilização do simulador por apenas um único aluno. Essa forma de uso é ideal para momentos de atividades individuais, em que o estudante necessita realizar a simulação e observar o resultado de suas configurações em seu próprio *smartphone*.

Figura 30 – Situação de uso do simulador por um aluno para outros alunos.



Fonte – o autor.

Figura 31 – Situação de uso do simulador por um aluno para observação do resultado em seu próprio *smartphone*.



Fonte – o autor.

4.2.2.2 Página Web de simulação

A página Web de simulação foi desenvolvida utilizando HTML e JavaScript. Para enviar as mensagens que disparam os eventos na aplicação Android, é preciso seguir o modelo apresentado pelo Código-fonte 3. Nesse sentido, “key” é uma chave de acesso, de cinco caracteres, gerada pela aplicação Web e inserida na tela de entrada da aplicação móvel para identificar quem deve receber as mensagens enviadas pela simulação Web. O “message.id” é o identificador da mensagem enviada, necessário para especificar ao que ela se refere, se ao sensor de luminosidade ou ao acelerômetro, por exemplo. Por fim, “\$(message).val()” é a mensagem propriamente dita que se deseja passar para disparar uma ação na aplicação Android.

Código-fonte 3 – Envio de mensagens da aplicação Web para a móvel.

```
1 ws.send("key," + "message.id" + $(message).val());
```

2

Para gerenciar as mensagens recebidas da aplicação Android, a função “*onmessage*”, tal como em Código-fonte 4, deve ser definida na aplicação Web da simulação.

Código-fonte 4 – Função de tratamento das mensagens recebidas na página Web de simulação.

```

1      ws.onmessage = function(messages) {
2          //Definição do que será feito com as mensagens
recebidas
3          };
4

```

4.2.2.3 Lucy móvel

O Lucy móvel é uma aplicação feita em Android para apresentar o resultado de todos os parâmetros configurados na tela de simulação Web. A versão Android do Lucy não é o próprio ambiente em versão *mobile*, mas sim uma extensão dele, uma espécie de observatório no qual se pode enxergar o resultado de um experimento feito.

Essa aplicação Android precisa conter todas as informações necessárias para que seja possível a observação dos eventos configurados na simulação Web planejada, incluindo o acesso aos sensores do *smartphone*. Para tratar as mensagens recebidas da simulação Web, a aplicação Android também precisa definir a função “*onMessage*” tal como no Código-fonte 5.

Código-fonte 5 – Tratamento das mensagens recebidas na aplicação Android.

```

1      public void onMessage(String s) {
2          final String message = s;
3          runOnUiThread(new Runnable() {
4              @Override
5              public void run() {
6                  //Definição do que será feito com as
mensagens recebidas
7              }

```

```

8         });
9     }
10

```

Para enviar mensagens da aplicação Android para a página Web de simulação, caso esta exija, o Código-fonte 6 deverá fazer parte do código Android do Lucy móvel.

Código-fonte 6 – Envio de mensagens da aplicação Android para a página Web de simulação.

```

1         websocketClient.send("key," + "message.id" +
2         editText.getText().toString());

```

4.2.3 Practice

Practice é a seção de atividades de implementação de pequenas aplicações e protótipos rápidos relacionados ao conceito ubíquo estudado. Nesta área, então, há os *hiperlinks* que direcionam os usuários do ambiente Lucy para o *download* de códigos com estruturas preestabelecidas para que corrijam possíveis erros existentes e acrescentem mais componentes, funções e eventos conforme as exigências da atividade desenvolvida. Além disso, pode-se encontrar questionários, guias para a realização da prática de simulação e exemplos de aplicações e códigos referentes aos temas estudados.

Assim como na seção de *Lecture*, nesta seção também é necessário acessar o código-fonte da página para disponibilizar os *hiperlinks* pretendidos. Uma tela de edição desses arquivos também deverá ser criada e disponibilizada para professores futuramente.

4.3 Conclusão

Este capítulo apresentou a proposta de um ambiente voltado para o ensino e aprendizagem de conceitos ubíquos, o Lucy. Com base nos estudos de trabalhos científicos sobre o ensino de Computação Ubíqua e nas pesquisas feitas com professores e alunos de diferentes instituições de ensino superior, foi possível propor esse ambiente com alguns dos requisitos e das características mencionados. Assim, o ambiente pode abordar diferentes conceitos ubíquos, com divisão entre áreas teóricas e práticas, simulação do funcionamento desses conceitos por

meio da utilização de sensores reais, integração de estruturas virtuais com dispositivos reais (*smartphones*) e implementações de aplicações e protótipos rápidos a partir de código com estruturas preestabelecidas disponibilizados no ambiente.

Um cenário motivador também foi apresentado a fim de ilustrar as possibilidades e maneiras de utilização do ambiente proposto para o ensino de conceitos ubíquos. Pela descrição do cenário, é possível notar as diferentes atividades que podem ser desenvolvidas para o ensino desses conceitos. Além disso, a abordagem na utilização da simulação pode ser de três modos: o professor usando a simulação Web para refletir nos dispositivos de todos os alunos, um pequeno grupo de alunos utilizando uma única página Web de simulação para refletir em seus aparelhos e cada aluno individualmente realizando uma atividade de simulação para observar o resultado em seu próprio celular.

O próximo capítulo traz a descrição da implementação das seções relativas ao tópico de Sensibilidade ao Contexto como prova de conceito do ambiente Lucy.

5 PROVA DE CONCEITO

Neste capítulo, apresenta-se o desenvolvimento de uma prova de conceito da ferramenta Lucy que implementa um módulo de Sensibilidade ao Contexto como tema integrante do ambiente Lucy. A Seção 5.1 explicita o motivo da escolha deste tema e a abordagem dele pela simulação do ambiente. A Seção 5.2 traz toda a estrutura do conceito de Sensibilidade ao Contexto criado dentro do Lucy, mostrando também seu funcionamento e exemplos das atividades desenvolvidas. A Seção 5.3 descreve as possibilidades e limitações de edição no conteúdo do Lucy. A Seção 5.4 expõe como o Lucy pode ser utilizado para aulas práticas de Sensibilidade ao Contexto. Por fim, a Seção 5.5 discute as conclusões a respeito da proposta desenvolvida apresentada neste capítulo.

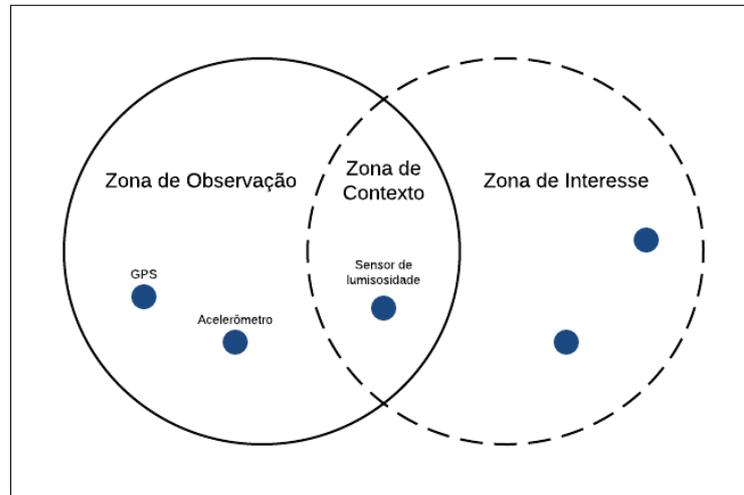
5.1 Sensibilidade ao Contexto

Identificado por muitos professores e alunos pesquisados como um dos conceitos mais importantes da Computação Ubíqua, Sensibilidade ao Contexto foi o primeiro tema a ser implementado dentro do ambiente Lucy. Desta forma, visa-se auxiliar os estudantes na experimentação desse conceito, mostrando o fluxo de dados em camadas e o comportamento de um sistema sensível ao contexto.

Na simulação desenvolvida para o tema de Sensibilidade ao Contexto, existem regras configuráveis por professores e alunos que mudam a zona de interesse contextual da aplicação. Em princípio, a ideia ilustra o conceito de contexto proposto por Viana (VIANA, 2010) (descrito em detalhes na Fundamentação Teórica, no Capítulo 2). Um conjunto de sensores está implementado na ferramenta e compõe a zona de observação do sistema. Com isso, as informações provenientes do GPS, do sensor de luz e do acelerômetro do dispositivo estão disponíveis para serem habilitadas e desabilitadas. Dependendo das configurações feitas pelo estudante para a zona de interesse, o contexto capturado pelo aplicativo móvel pode estar relacionado apenas às informações fornecidas pelo sensor de luz, como mostrado na Figura 32.

A simulação desenvolvida faz também uma analogia com as camadas de um sistema sensível ao contexto com base no *Context Toolkit framework* proposto por Dey (DEY; ABOWD, 1999), que está ilustrado na Figura 33, e na estrutura apresentada por Coutaz (COUTAZ *et al.*, 2005), mostrada na Figura 34. As abstrações realizadas para a construção destas arquiteturas favorecem o entendimento de conceitos para a construção de aplicações sensíveis ao contexto.

Figura 32 – Exemplificação de contexto para a simulação de Lucy seguindo as definições de Viana (VIANA, 2010).

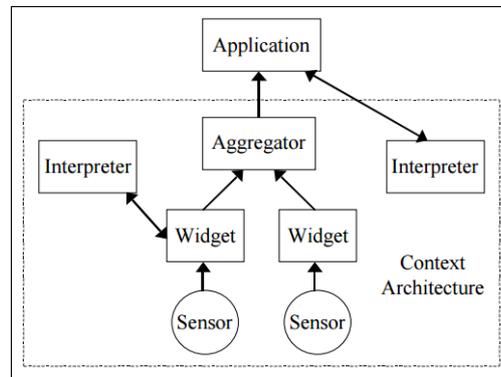


Fonte – o autor.

Na estrutura proposta por este trabalho, 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. Tem-se ainda os interpretadores, que interpretam informações de contexto de baixo nível, como o valor de luminosidade em lx, 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. Por último, tem-se a tomada de decisão, que identifica o contexto de uma entidade para disparar um evento ou uma ação.

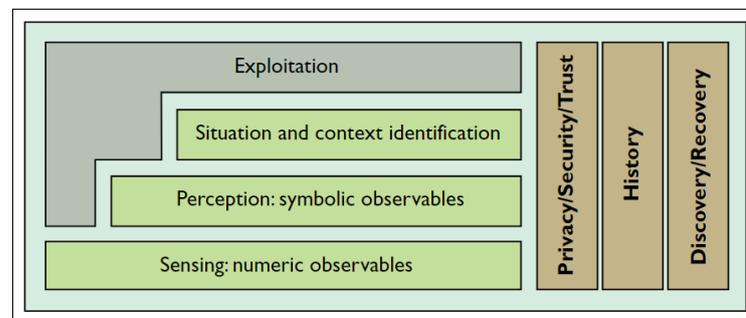
A importância de estudar as arquiteturas de (DEY; ABOWD, 1999) e (COUTAZ *et al.*, 2005) se dá pela compreensão de como funciona e se comporta o *background* dos sistemas sensíveis ao contexto numa estrutura de poucas camadas. Além disso, há vários trabalhos desenvolvidos com base nessas arquiteturas, o que as tornam ainda mais indispensáveis ao estudo. Como exemplo de alguns desses trabalhos, pode-se citar (VEIGA *et al.*, 2014), (DUARTE *et al.*, 2015), (FERREIRA *et al.*, 2015) e (WENDT; JULIEN, 2016).

Figura 33 – Estrutura do *Context Toolkit framework* proposto por Dey.



Fonte – (DEY; ABOWD, 1999)

Figura 34 – Arquitetura proposta por Coutaz.



Fonte – (COUTAZ *et al.*, 2005)

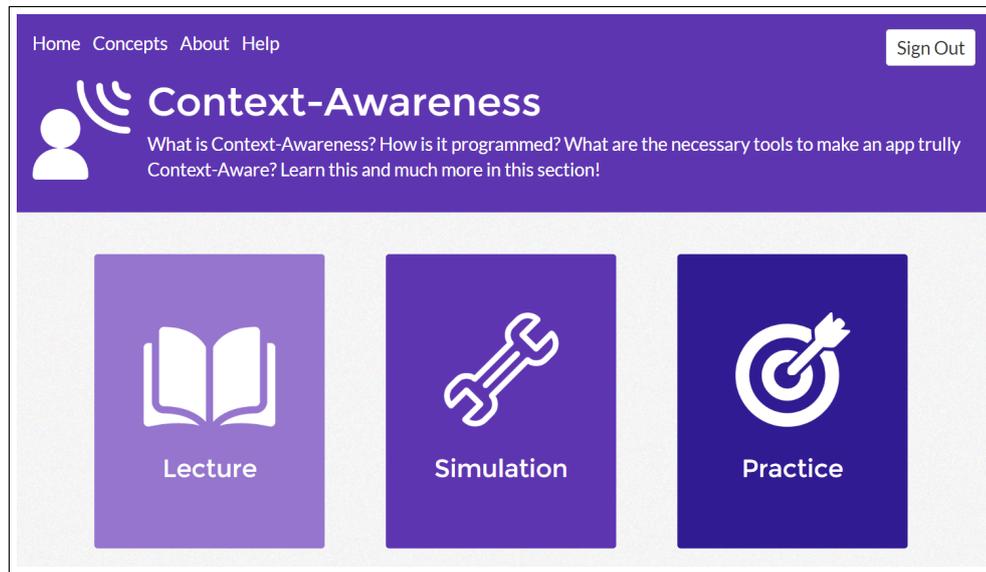
5.2 Implementação no ambiente Lucy

No ambiente Lucy, o conceito de Sensibilidade ao Contexto possui três seções disponíveis: *Lecture*, *Simulation* e *Practice*. A Figura 35 ilustra essa divisão. Cada uma dessas seções traz materiais específicos utilizados em sala de aula pelo professor em suas atividades teóricas e práticas.

5.2.1 Lecture

Na seção de *Lecture* de Sensibilidade ao Contexto, há uma explicação sobre o *framework* usado como base para desenvolver as atividades relacionadas a esse tema. Existem também *links* que levam ao trabalho dos autores responsáveis pela construção do *framework*, o *Context Toolkit*. A Figura 36 apresenta um exemplo de conteúdo que pode ser exibido na seção de *Lecture* de Sensibilidade ao Contexto. O material disponibilizado nesta seção, mostrado pela Figura 36, representa apenas uma amostra do que é possível ser usado como complemento ou auxílio de aula teórica e também como apoio para sanar algumas dúvidas dos estudantes.

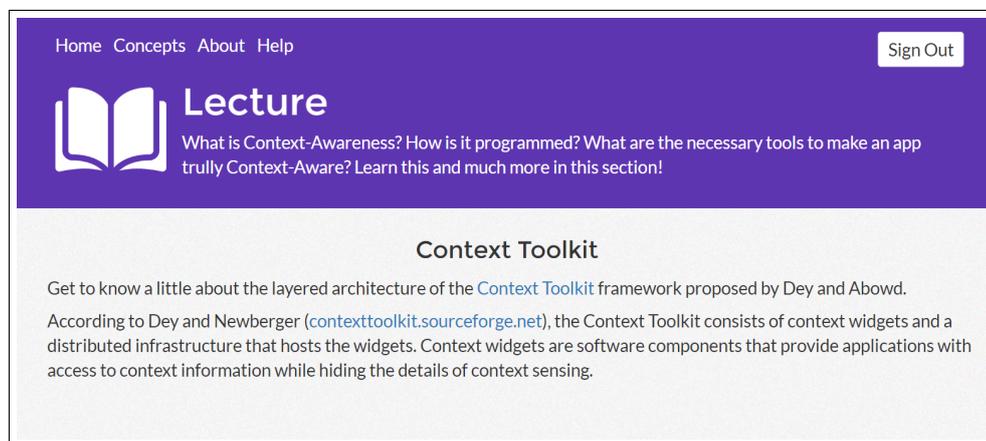
Figura 35 – Divisão do conceito de Sensibilidade ao Contexto dentro do Lucy.



Fonte – o autor.

O professor pode ainda inserir outros artigos e projetos ligados ao tema e dividir a seção em materiais principal e complementar.

Figura 36 – Seção de *Lecture* de Sensibilidade ao Contexto.



Fonte – o autor.

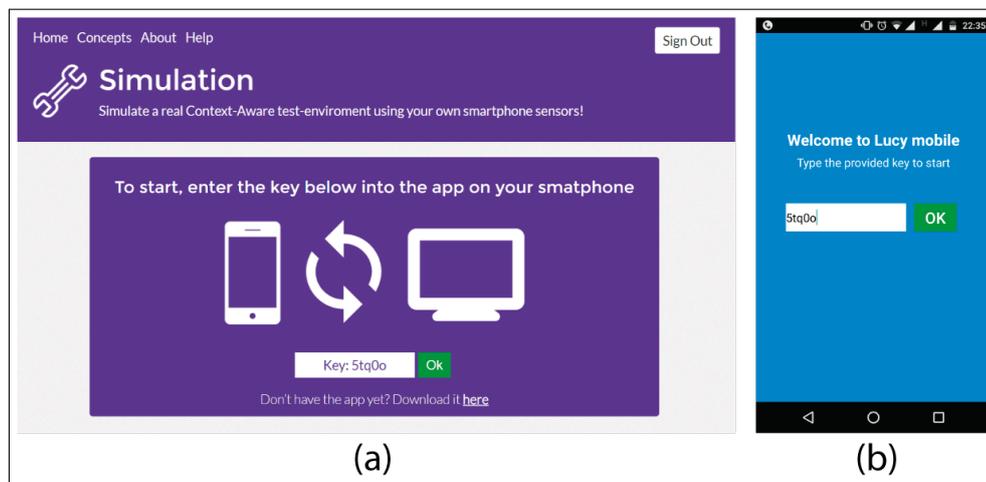
5.2.2 *Simulation*

A seção de simulação deste conceito apresenta uma estrutura em camadas que 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

(CARVALHO *et al.*, 2011).

Essa simulação então é dividida em duas partes: o lado Web, que é responsável pela configuração de informações e criação de regras, e o lado móvel, que tem a função de apresentar o resultado das configurações feitas na interface Web. Para conectar a aplicação móvel à interface Web de simulação, basta inserir a chave de acesso mostrada na Figura 37 (a) dentro da caixa de texto exibida pela Figura 37 (b).

Figura 37 – Conexão da aplicação móvel com a interface Web de simulação do Lucy.



Fonte – o autor.

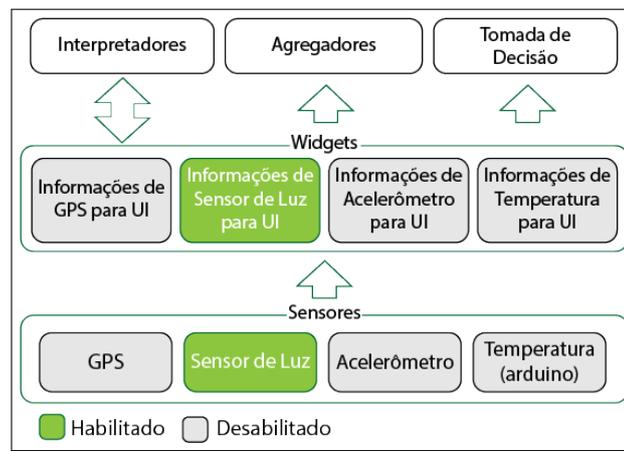
5.2.2.1 Interface Web de simulação

A interface Web de simulação traz uma estrutura em camadas, onde sensores podem ser habilitados, desabilitados e configurados de acordo com os objetivos do usuário, além da inserção de parâmetros e de tomadas de decisão presentes nesta estrutura. Nessa parte do ambiente, novas regras contextuais também podem ser criadas. Todas as configurações realizadas na simulação Web podem ser visualizadas e testadas na aplicação móvel do Lucy. No ambiente proposto, conforme a Figura 38, 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. Vale ressaltar que o conceito de camadas adotado não é exatamente o que está descrito no *framework* proposto por Dey (DEY; ABOARD, 1999). Ele é um misto do que é proposto no *Context Toolkit* com o que foi descrito em Coutaz (COUTAZ *et al.*, 2005).

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 um sensor de temperatura ambiente rodando em Arduino. O sensor de temperatura não está implementado. Ele ficou totalmente desabilitado, somente indicando o local do próximo sensor a ser implementado, com o intuito de instigar a curiosidade e o debate entre os estudantes. Inicialmente, o GPS captura a localização atual do usuário e, então, essa localização pode ser modificada por meio da inserção de novas coordenadas na interface de simulação Web.

Figura 38 – Esquema da tela da simulação sobre Sensibilidade ao Contexto.



Fonte – o autor.

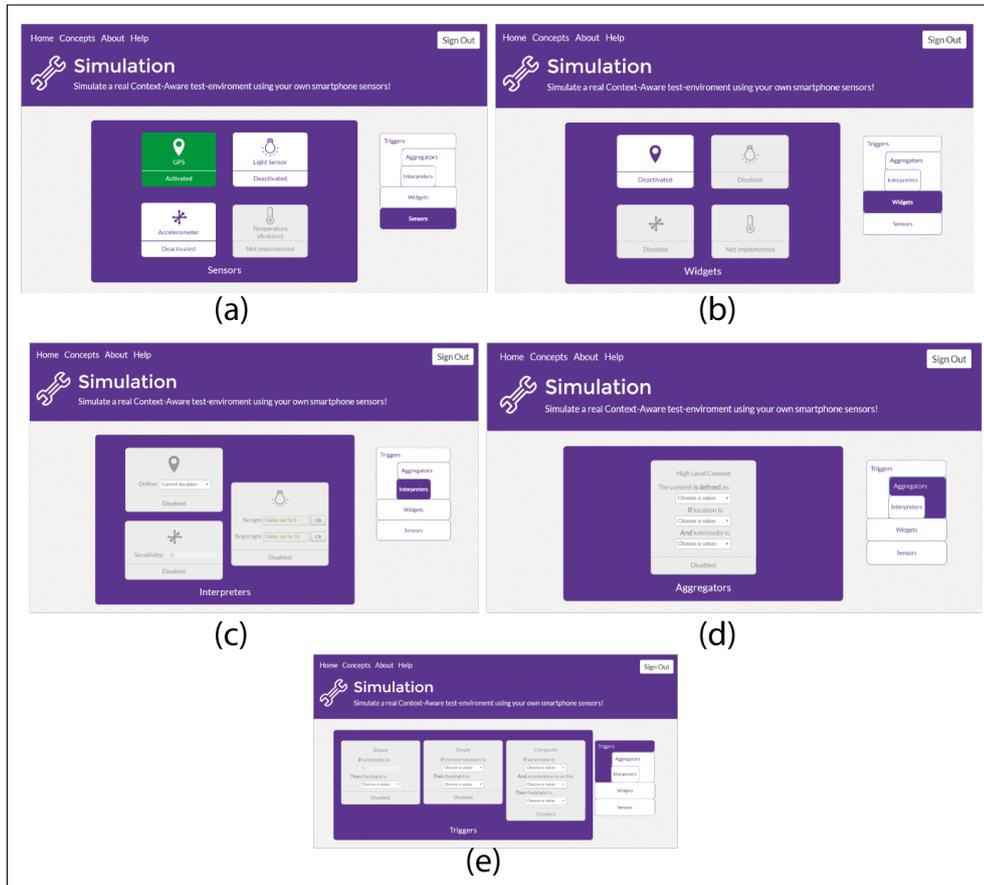
A camada de *Widgets* é responsável por mostrar, na aplicação móvel do usuário, dados adquiridos pelo sensor habilitado. Os agregadores juntam dados dos sensores para gerar uma nova informação contextual de alto nível. A camada de interpretadores interpreta dados de baixo nível e apresenta uma informação de mais alto nível. Na Tomada de Decisão ou *Triggers*, tanto podem ser utilizados os dados das camadas de *Widgets* e de interpretadores como as informações contextuais geradas pelos agregadores, um componente do celular (e.g., lanterna) é ativado ou desativado conforme a captura de informações configuradas pelo usuário na simulação.

A Figura 39 exibe a estrutura da simulação criada para o conceito de Sensibilidade ao Contexto implementada dentro do ambiente Lucy.

5.2.2.2 Lucy móvel

O Lucy móvel foi criado em versão Android. Essa tecnologia foi escolhida por ser gratuita e bastante utilizada por alguns professores em suas aulas práticas. Nessa parte do ambiente, observa-se o comportamento de um sistema sensível ao contexto conforme as

Figura 39 – Esquema da simulação de Sensibilidade ao Contexto. (a) Camada de sensores. (b) Camada de *Widgets*. (c) Camada de interpretadores. (d) Camada de agregadores. (e) Camada de *Triggers*.

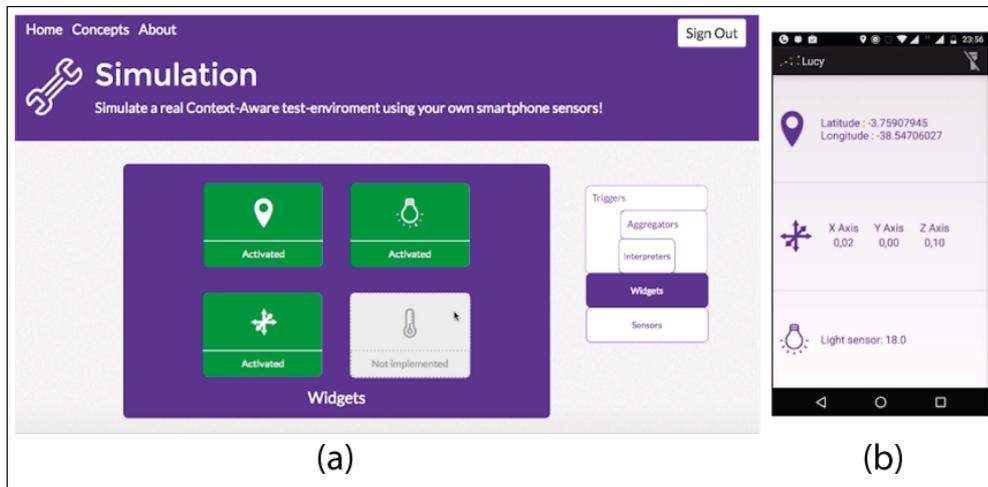


Fonte – o autor.

configurações feitas no ambiente Web pelo professor ou pelos alunos. O sistema móvel não apresenta, inicialmente, nenhuma informação na área dos sensores (Figura 41 (a)), que vai sendo preenchida conforme as configurações dos usuários na interface Web de simulação. Assim, se o sensor de luminosidade for habilitado dentro da camada de *widgets* no sistema Web, conforme mostra a Figura 40 (a), valores em lx aparecem no sistema móvel. Esses valores mudam de acordo com a variação de intensidade da luz no ambiente real captada pelo sensor do celular como ilustrado na Figura 40 (b).

A aplicação móvel só apresenta as informações configuradas pelo usuário no ambiente Web à medida que elas vão sendo adquiridas do contexto desse usuário. O sistema apresenta, conforme mostra a Figura 41 (b), a informação de sensores habilitados para a aplicação. A Figura 41 (c) exhibe um valor de luminosidade em lx obtido a partir do sensor do próprio dispositivo, dados do acelerômetro e as coordenadas de localização do usuário. Tem-se também, conforme a

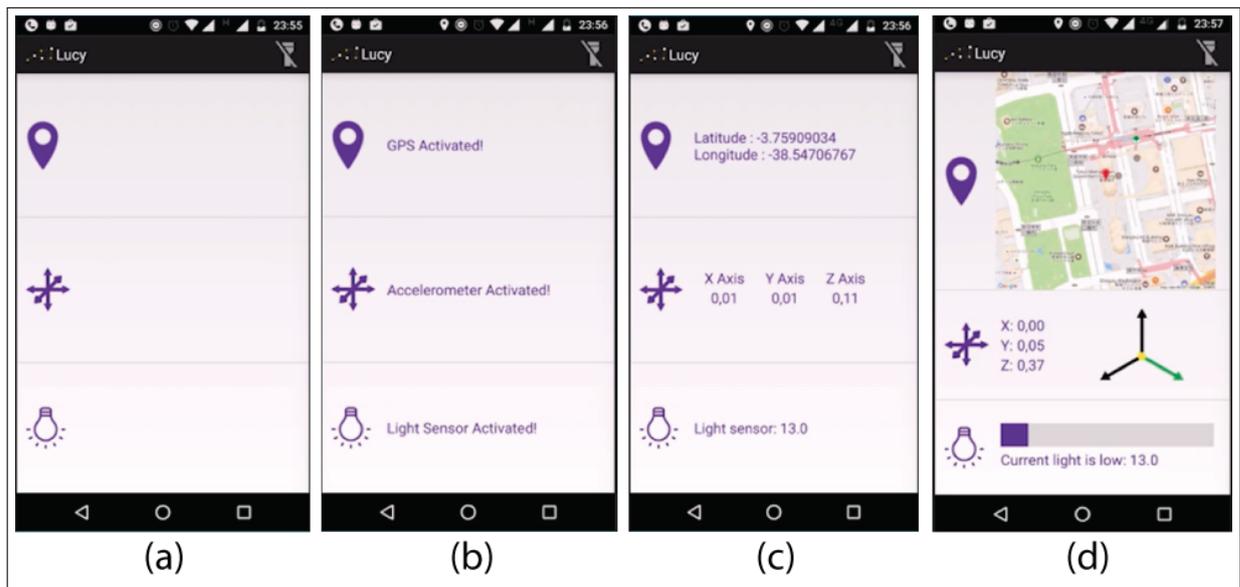
Figura 40 – Ativação das informações dos sensores na camada de *widjets*.



Fonte – o autor.

Figura 41 (d), 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*”. A tela também exibe uma figura que interpreta o acelerômetro que apresenta variação nos valores dos eixos X, Y e Z à medida que o dispositivo é chacoalhado e o mapa da localização do usuário.

Figura 41 – Telas do Lucy móvel. (a) Área inicial dos sensores. (b) Sensores habilitados. (c) Dados dos sensores na camada de *widjets*. (d) Dados dos sensores na camada de interpretadores.

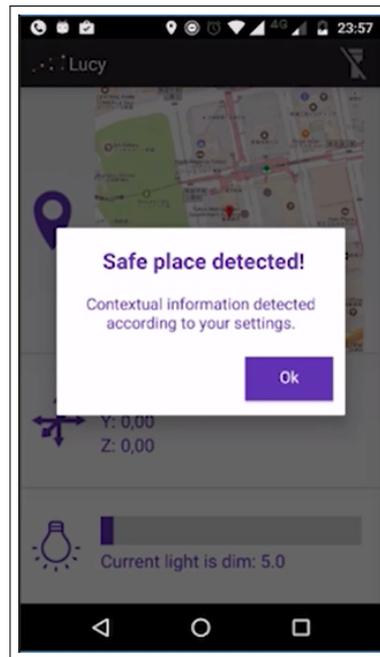


Fonte – o autor.

Na Figura 42, tem-se um alerta indicando que, segundo as configurações que o

usuário fez na página de simulação 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 simulação.

Figura 42 – Reconhecimento de contexto seguro.



Fonte – o autor.

5.2.2.3 Exemplo de atividade com a simulação

O objetivo da atividade de simulação é mostrar/simular o fluxo de informações em aplicações sensíveis ao contexto, proporcionando aos alunos uma noção do que acontece no *background* desse tipo de aplicação. A simulação utilizada possui camadas de sensores, *widjets*, interpretadores, agregadores e tomadas de decisão. Cada uma dessas camadas, com suas características específicas, segue princípios dos componentes propostos por Dey e Coutaz.

Na atividade prática com a simulação de Sensibilidade ao Contexto, os alunos acessam o documento mostrado no Apêndice D, que está na seção de *Practice* do conceito estudado, e seguem as instruções fornecidas. Os estudantes precisam responder às solicitações feitas no documento e incluir neste imagens das configurações feitas na interface de simulação Web e dos resultados obtidos através do dispositivo móvel.

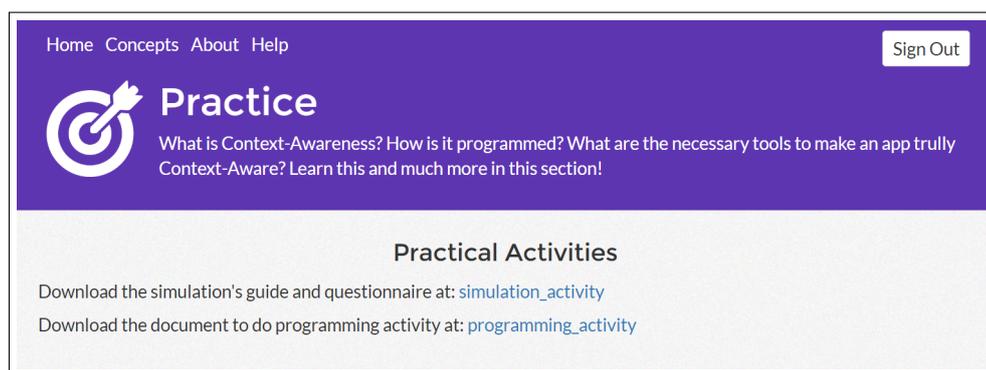
Exemplos do que é pedido pelo documento que guia a atividade de simulação, pode

ser ilustrado como: “Habilite o sensor de luminosidade e seu *Widget* correspondente. Descreva o que aconteceu. Em seguida, tente alterar o valor da luminosidade para zero.” e “Habilite um agregador de contexto e modifique seus parâmetros para definir o contexto de alto nível “Perigoso”. O que aconteceu na aplicação móvel? Modifique o seu contexto atual para que o contexto de alto nível Perigoso seja produzido.”

5.2.3 *Practice*

Na área denominada *Practice*, encontram-se os *links* para guias e questionários que orientam tanto a simulação quando a atividade prática de programação. A Figura 43 apresenta o conteúdo referente a essa seção do conceito de Sensibilidade ao Contexto. Na atividade prática de programação, os estudantes recebem uma estrutura de código Android semelhante à estrutura da simulação de Sensibilidade ao Contexto, com as mesmas camadas que a compõem e que fazem parte do *Context Toolkit framework* de Dey. Esse código Android possui dois sensores, sensor de luz e acelerômetro, com suas respectivas informações distribuídas pelas camadas presentes no código. Os alunos precisam compilar esse código que receberam, corrigir dois erros no funcionamento da aplicação resultante e também acrescentar pelo menos um sensor e suas informações nas camadas de *widgets*, interpretadores, agregadores e *triggers*. A Figura 44 mostra um diagrama de classes do código Android destinado à prática de programação.

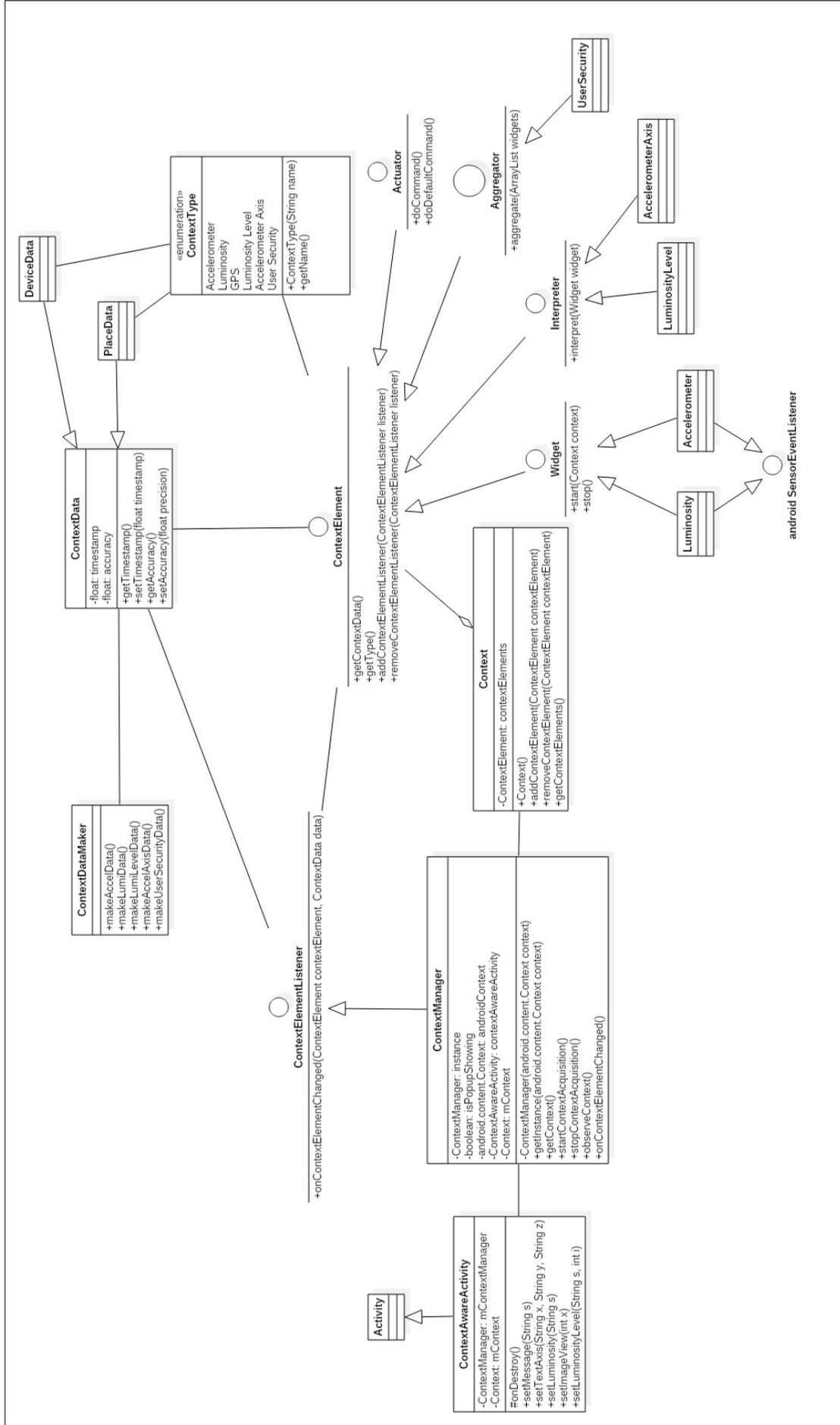
Figura 43 – Seção de *Practice* de Sensibilidade ao Contexto.



Fonte – o autor.

Para a prática de programação, os alunos devem acessar o documento mostrado no Apêndice F, que está na seção de *Practice* do conceito estudado, e se guiar por ele para realizar toda a atividade. Como exemplo do que é pedido nesse documento, pode-se observar alguns trechos dele: “No código inicial proposto pela prática, quais os elementos contextuais existentes?”

Figura 44 – Diagrama de classes do código Android referente à prática de programação.



Quais atributos dessas classes poderiam ser definidos como propriedades contextuais de baixo nível (i.e., providas por sensores ou *widgets*) e quais seriam de alto nível (i.e., providas por agregadores e/ou interpretadores)? Explique.” e “Acrescente um novo sensor à aplicação e crie seu correspondente *widget*, interpretador, agregador e uma tomada de decisão com um atuador.”.

O objetivo dessa atividade prática de programação é proporcionar aos alunos a manipulação de uma estrutura de código de aplicação sensível ao contexto seguindo a arquitetura proposta pela simulação do Lucy. Isso para que os estudantes consigam perceber e entender a codificação da estrutura de simulação trabalhada. Além disso, há a possibilidade de criação de aplicações de forma simplificada, pois os alunos recebem um projeto Android com todas as camadas presentes na simulação, tendo então que modificar os sensores, as agregações e as tomadas de decisão dependendo da aplicação que queiram criar.

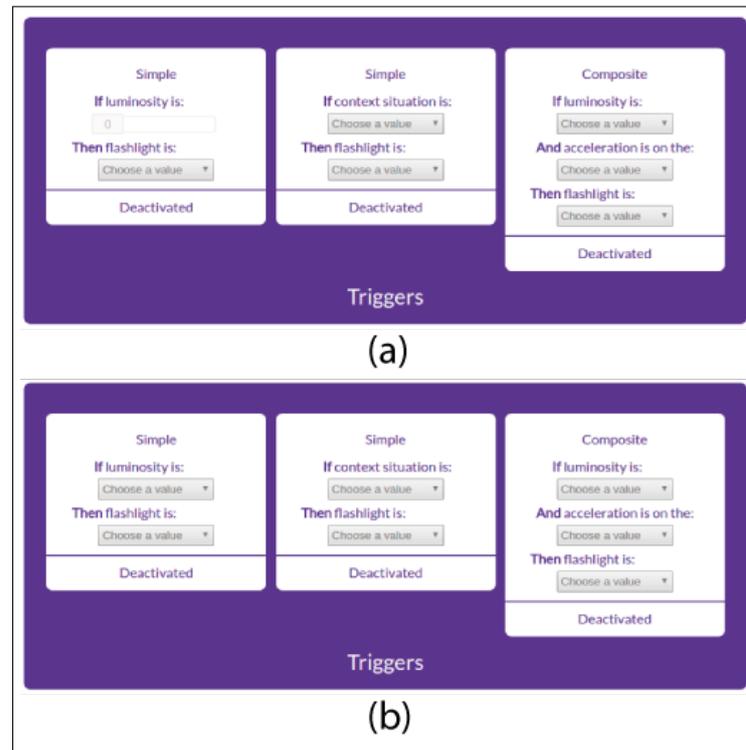
5.3 Edição e Customização

Em testes de avaliação do ambiente com professores, observou-se a necessidade de realização de uma espécie de customização na área de simulação. Alguns professores preferiram utilizar os valores de luminosidade da primeira caixa da camada de *Triggers* em intervalos (e.g., “*no light*”, “*dim*”, “*low*” e “*bright*”) e outros quiseram que esses valores fossem números inteiros.

Inicialmente, fez-se esses valores de referência da variação de luminosidade com números inteiros, pois objetivou-se exemplificar para os alunos que a camada de *Triggers* poderia receber valores das camadas de *Widgets*, interpretadores e agregadores. Assim, haviam três caixas alimentadas por dados de cada uma dessas camadas, e a primeira caixa recebia dados mais brutos provenientes da camada de *widgets*, logo, números inteiros. No entanto, alguns professores preferiam os dados de luminosidade dessa primeira caixa divididos em intervalos numéricos. Com isso, fez-se uma alteração no código da simulação para se ter as duas possibilidades a escolha do professor, para que, caso ele quisesse valores divididos em intervalos, fosse comentado o código referente aos valores inteiros.

Desta forma, existe a viabilidade de edição na simulação, embora que ainda no código, para que se possa ter a primeira caixa da camada de *Triggers* com os valores de luminosidade apresentados conforme a escolha do professor, dentro das alternativas oferecidas. A Figura 45 mostra as duas possibilidades da caixa. Na Figura 45 (a), tem-se os valores de luminosidade apresentados como números inteiros e, na Figura 45 (b), tem-se a divisão desses valores em intervalos.

Figura 45 – Possibilidades diferentes de representação dos valores de luminosidade da primeira caixa da camada de *triggers*. (a) Valores de luminosidade representados por números inteiros. (b) Valores de luminosidade representados por intervalos (e.g., “no light”, “dim”, “low” e “bright”).



Fonte – o autor.

O ideal é que essas mudanças de configuração pudessem ser feitas sem a necessidade de alterações no código. Na verdade, telas de edição para todas as seções do conceito de Sensibilidade ao Contexto deveriam ser implementadas, mas não foram ainda em virtude de limitações de tempo.

5.4 Sugestão de uso

Nesta seção, apresenta-se uma sugestão de uso do módulo implementado. Foi essa a metodologia de uso do recurso adotada nos experimentos expostos no Capítulo 6.

Para utilizar o Lucy em aulas referentes ao conceito de Sensibilidade ao Contexto, primeiro o professor ministra uma aula teórica sobre o tema. Tem-se a necessidade de abordar características e definições desse conceito, assim como exemplos de aplicações, nessas aulas expositivas.

Na aula seguinte a essa teórica, o professor faz uma aula teórica-prática utilizando a

simulação do Lucy. Assim, todos os alunos baixam a aplicação móvel do Lucy em seus próprios celulares e se conectam à interface Web de simulação aberta pelo professor em seu computador e visível para todos na sala. Feito isso, o professor começa a configurar a simulação explicando os objetivos pretendidos. Então, as configurações feitas pelo professor na interface Web de simulação são refletidas em todos os celulares conectados a ela. Em seguida, o professor orienta os estudantes a se cadastrarem no ambiente Lucy e seguir o guia do Apêndice D da simulação que está disponível na área de *Practice* do conceito de Sensibilidade ao Contexto.

Na aula posterior à prática com a simulação, realiza-se então uma atividade prática de programação. Para esta atividade, os estudantes são orientados pelo professor a acessar o Lucy e obter o documento de orientação (Apêndice F) que está disponível na seção de *Practice* do conceito estudado. Junto com o guia de orientação do exercício de programação, os alunos também acessam um arquivo com código Android que possui as mesmas camadas da simulação nele definidas. Com esse código Android, os estudantes precisam realizar todas as solicitações pedidas no guia de orientação (Apêndice F).

5.5 Conclusão

Este capítulo apresentou a implementação do conceito de Sensibilidade ao Contexto dentro do ambiente Lucy. Este tema, com seções de *Lecture*, *Simulation* e *Practice*, foi construído como prova de conceito para o Lucy. Foram apresentadas as características e definições do tema ubíquo estudado, a estrutura deste, o funcionamento, os tipos de atividades que podem ser desenvolvidas com ele e a forma de utilização dele em aulas de Sensibilidade ao Contexto.

O conceito de Sensibilidade ao Contexto foi implementado dentro do Lucy utilizando a estrutura proposta pelo ambiente, tais como a divisão do conceito em seções de teoria, simulação e prática, assim como a comunicação da simulação com *smartphones* e a utilização dos sensores destes pela simulação.

O próximo capítulo trata da exposição dos resultados das avaliações realizadas com o ambiente em diferentes turmas e semestres de disciplinas relacionadas à Computação Ubíqua. Houve também avaliação do Lucy com professores que trabalham os conceitos ubíquos em suas instituições de ensino superior. Então, estes puderam mensurar a validade e a importância do ambiente para aulas práticas de conceitos ubíquos na graduação e pós-graduação.

6 AVALIAÇÃO

Neste capítulo, são relatadas as avaliações de usabilidade e de interface do Lucy Web e móvel realizadas com professores e alunos de disciplinas ligadas à Computação Ubíqua. Além dessas, houve algumas perguntas com o intuito de perceber aspectos pedagógicos. Foram também feitas, com os estudantes, as avaliações de conhecimentos teóricos antes e depois das atividades de simulação do Lucy e uma avaliação sobre a prática de programação proposta pelo ambiente. Com os professores, foi ainda realizada uma avaliação com o objetivo de conhecer a experiência dele relacionada à Computação Ubíqua e a sua satisfação ao conhecer e utilizar o Lucy e os seus materiais.

A Seção 6.1 apresenta as avaliações realizadas e os resultados obtidos com os estudantes de uma disciplina relacionada à Computação Ubíqua. A Seção 6.2 descreve as avaliações e os resultados dos testes com o Lucy feitos com seis professores de diferentes universidades. Por fim, a Seção 6.3 traz as conclusões relativas às avaliações do Lucy.

6.1 Avaliação com estudantes

A finalidade desta seção é relatar o processo avaliativo realizado com os alunos de disciplinas relacionadas à Computação Ubíqua, assim como os resultados preliminares obtidos. No processo de composição do ambiente e de suas funcionalidades, foram sendo realizados testes com os usuários finais, professores e alunos, conforme definem as orientações da metodologia do DCU. Com isso, três avaliações do ambiente foram realizadas em aulas práticas de Sensibilidade ao Contexto.

Essas avaliações foram feitas em três momentos com 20 alunos da UFC. No primeiro momento, 6 estudantes de graduação da disciplina de Introdução à Computação Móvel e Ubíqua 2016.1 usaram o Lucy. No segundo momento, 8 estudantes de graduação e de pós-graduação da disciplina de Computação Móvel e Ubíqua 2016.2 participaram da avaliação. No terceiro momento e último teste feito, 6 estudantes de graduação da disciplina de Introdução à Computação Móvel e Ubíqua 2016.2 fizeram uso do ambiente.

Nos dois primeiros momentos, as avaliações de usabilidade e de interface não foram feitas da forma detalhada como na última avaliação. Nesses dois primeiros testes, foi aplicado um questionário baseado no trabalho de (REATEQUI *et al.*, 2010) e que possuía questões que tentavam mensurar problemas de funcionamento, interface e pedagógico.

Assim, no primeiro teste, 87,5% dos estudantes apontaram algum tipo de erro na simulação. Esse valor caiu para 66,7% no segundo teste. Esses erros eram, em geral, com relação a falhas na exibição das informações no *smartphone*, à falta da possibilidade de desabilitar as tomadas de decisão e a paradas inesperadas do sistema. Esses erros foram sendo corrigidos e, no terceiro teste, os alunos não relataram mais falhas que pudessem comprometer o uso do ambiente ou a realização das atividades estabelecidas.

Outro ponto considerado foi que, já no primeiro teste, 62,5% dos alunos disseram que o Lucy instigava a procura de outras informações em diferentes fontes de pesquisa. Esse valor aumentou para 100% na realização da segunda avaliação.

Conforme os testes foram sendo feitos com o Lucy, as modificações necessárias, como os exemplos citados acima, também foram sendo observadas e realizadas até que se chegasse a uma estrutura mais estável do ambiente, que atendesse às necessidades de estudantes e professores. Para a discussão das avaliações e dos resultados obtidos com os alunos, são levados em consideração apenas os dados do último teste, pois este foi feito com a versão mais estável do Lucy, a qual possui implementadas as alterações observadas em avaliações anteriores.

6.1.1 Perfil dos alunos

O grupo dos avaliados era composto por 6 estudantes de graduação da disciplina de Introdução à Computação Móvel e Ubíqua do curso de Sistemas e Mídias Digitais da UFC. Todos os participantes deste teste eram do sexo masculino e estavam em semestres diferentes do curso.

6.1.2 Materiais e métodos

Para esta última avaliação, o ambiente Lucy estava online e poderia ser acessado por navegadores Web, tais como Mozilla Firefox, Google Chrome ou Internet Explorer. O *smartphone* utilizado para a observação dos resultados da simulação deveria executar o sistema Android 4.0 ou superior, sendo utilizado os dispositivos dos próprios participantes da avaliação.

Para verificar a capacidade de definição de alguns conceitos teóricos relacionados à Sensibilidade ao Contexto, foi aplicado o questionário do Apêndice C antes e depois das atividades de simulação. O objetivo desse questionário era comparar as diferenças nas definições desses conceitos antes e depois da realização das atividades de simulação.

Outro questionário, Apêndice E, foi aplicado após a finalização das atividades de

simulação. O questionário do Apêndice E foi dividido em 4 seções: avaliação de usabilidade, avaliação de aspectos pedagógicos, avaliação de interface e área para sugestões. A seção de avaliação de usabilidade foi baseada no System Usability Scale (SUS) (BROOKE, 2013), um método de averiguação do nível de usabilidade de um sistema. Esse método procura avaliar a eficácia (os usuários conseguem completar seus objetivos?), eficiência (quanto esforço e recursos são necessários para isso?) e satisfação (a experiência foi satisfatória?) do usuário através de 10 questões com respostas que podem variar de 1 a 5, em que 1 significa “Discordo Completamente” e 5 significa “Concordo Completamente”. Por ser um questionário curto e de respostas rápidas, o SUS estabelece que as questões sejam alternadas entre afirmações positivas e negativas a fim de se evitar viés nas respostas e fazer com que os pesquisados leiam atentamente cada declaração para responder se realmente concordam ou não (BROOKE, 2013). Ao final uma pontuação é conseguida conforme as regras estabelecidas pelo SUS e varia de 0 a 100. A pontuação média do SUS é de 68 pontos, então tudo considerado acima disso está acima da média. A escala que representa a variação dessa pontuação está dividida em “A”, “B”, “C”, “D” e “F”, na qual em “A” estão as aplicações com pontuação acima de 80 e em “F” aquelas com menos de 51 pontos (SAURO, 2011).

As seções de avaliação de aspectos pedagógicos e interface do questionário do Apêndice E foram inspiradas nos itens que compõem a metodologia PETESE (COOMANS; LACERDA, 2015), que propõe um referencial para a avaliação de *softwares*. O questionário, disponível no Apêndice E, contém afirmações e os alunos deveriam responder qual seu nível de concordância com aquela afirmação numa escala de 1 a 5, em que 1 significa “Discordo Completamente” e 5 significa “Concordo Completamente”.

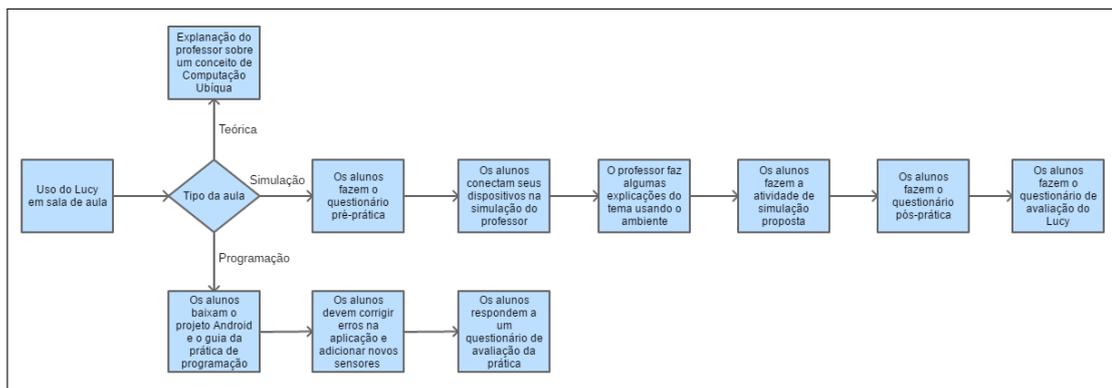
No final da aula referente à prática de programação, foi também aplicado o questionário do Apêndice F, cujo o objetivo era verificar a utilidade do formato e da aplicação da prática de programação. Os alunos tinham que responder a quatro itens. Dois destes deveriam ser respondidos com uma das alternativas que variam de 1 a 5, em que 1 significa “Discordo Completamente” e 5 significa “Concordo Completamente”. Os dois últimos itens eram discursivos e buscavam a opinião do aluno sobre a atividade proposta.

6.1.3 Procedimento

A utilização do Lucy em atividades práticas de Sensibilidade ao Contexto se deu em duas aulas seguidas. Antes disso, porém, o professor explanou sobre o tema em uma aula

teórica. A Figura 46 ilustra as etapas deste procedimento. Em cada uma dessas aulas práticas, foi disponibilizado o tempo de uma hora e trinta minutos para a realização das tarefas com o ambiente. Dentro desse tempo definido, a primeira aula prática foi destinada às atividades de simulação e às respostas aos questionários ligados a essa prática. Primeiramente, os estudantes responderam ao questionário pré-prática do Apêndice C. Em seguida, o professor iniciou a aula utilizando a simulação do Lucy com os dispositivos dos alunos conectados, a fim de exemplificar o funcionamento e as características do ambiente e do conceito estudado. Logo depois, os alunos realizaram a prática de simulação utilizando o guia do Apêndice D destinado a essa atividade. Por fim, os estudantes responderam ao questionário do Apêndice E construído para a avaliação do ambiente e também ao questionário do Apêndice C pós-prática.

Figura 46 – Etapas do procedimento realizado nas aulas com o Lucy.



Fonte – o autor.

Na aula seguinte, os alunos acessaram o menu de “*Practice*” do Lucy e baixaram o guia do Apêndice F e o código Android referentes à atividade prática de programação. Para este exercício, foi necessário executar o código fornecido pelo ambiente no programa Android Studio e seguir o que era pedido no guia do Apêndice F para completar a tarefa. Ao final desta aula, os estudantes responderam ao questionário do Apêndice G para avaliação da prática de programação. A execução dos trabalhos dessa aula também durou uma hora e trinta minutos.

6.1.4 Resultados

Com o questionário de comparação das definições teóricas (Apêndice C), observa-se que as respostas dos alunos estão mais bem definidas e detalhadas no momento posterior à realização das atividades de simulação. O Quadro 6 traz um exemplo das respostas dadas pelos estudantes a uma das questões do questionário do Apêndice C. Apenas o aluno 5 conseguiu uma

resposta mais coerente para a pergunta realizada no momento pré-prática.

Quadro 6 – Comparativo das respostas dos estudantes a uma das questões do questionário do Apêndice C.

	Pré-prática	Pós-prática
	Como você pode diferenciar os <i>widgets</i> de interpretadores?	
Aluno 1	Os sensores diferentes associados a eles.	Os <i>widgets</i> mostram informações de baixo nível, mas que através dos interpretadores podem ser facilmente traduzidas a informações de alto nível.
Aluno 2	Os <i>widgets</i> recebem as informações dos sensores (como nível de ruído) e as transformam em informações importantes para a aplicação (começou uma reunião na sala).	<i>Widgets</i> são interfaces para visualização de dados de sensores. Interpretadores utilizam esses dados para derivar uma informação (ex.: pouca luz, muita luz).
Aluno 3	O primeiro seria mais sofisticado.	Os <i>widgets</i> mostram informações brutas sobre o senso em específico, enquanto os interpretadores refinam essa informação e dão um <i>feedback</i> rapidamente reconhecido.
Aluno 4	Não sei responder.	Os <i>widgets</i> são interfaces para as verdadeiras fontes de dados, que ficam na camada abaixo. Os interpretadores utilizam essas interfaces e tentam obter algum significado dos dados puros.
Aluno 5	<i>Widgets</i> retornam dados dos sensores para serem usados na definição de um contexto. Interpretadores sintetizam dados e geram informações de nível mais alto a respeito de um contexto.	<i>Widgets</i> possibilitam a comunicação da aplicação com os sensores do dispositivo. Interpretadores usam os dados providos pelos sensores para gerar informações de alto nível relevantes para a aplicação.
Aluno 6	Interpretadores retornam o valor de sensores ou uma informação que é feita com a análise de vários deles. O <i>widgets</i> são grupos de interpretadores que fornecem dados para a aplicação.	<i>Widgets</i> são a comunicação dos sensores com o sistema, através deles obtemos os valores brutos dos sensores. Os interpretadores agem em cima dos dados dos <i>widgets</i> , definindo estados e informações mais contextuais.

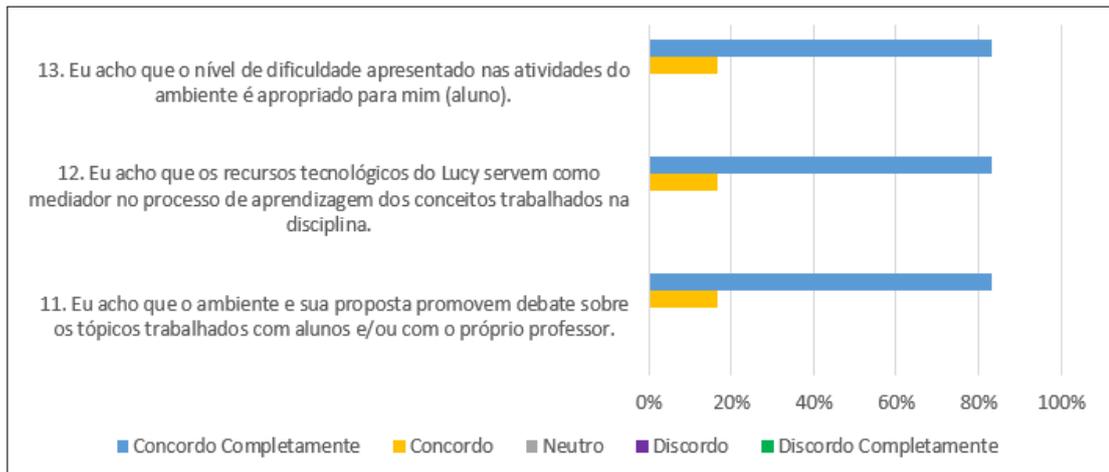
Fonte – o autor.

No questionário do Apêndice E, os estudantes avaliaram a usabilidade do sistema como satisfatória, mensurada pela pontuação de 75,8 pontos obtidos de acordo com os procedimentos do SUS (BROOKE, 2013), o que indica um *software* com boa usabilidade. Segundo a avaliação dos alunos, o Lucy está na faixa “B”, que comporta as aplicações com a pontuação de usabilidade maior que 70 e menor ou igual a 80 pontos.

Nas três afirmações da avaliação de aspectos pedagógicos, conforme a Figura 47 mostra, 83% dos estudantes responderam que concordam completamente com: “Eu acho que o ambiente e sua proposta promovem debate sobre os tópicos trabalhados com alunos e/ou com o

próprio professor”, “Eu acho que os recursos tecnológicos do Lucy servem como mediador no processo de aprendizagem dos conceitos trabalhados na disciplina” e “Eu acho que o nível de dificuldade apresentado nas atividades do ambiente é apropriado para mim (aluno)”.

Figura 47 – Respostas da avaliação de aspectos pedagógicos dos alunos.

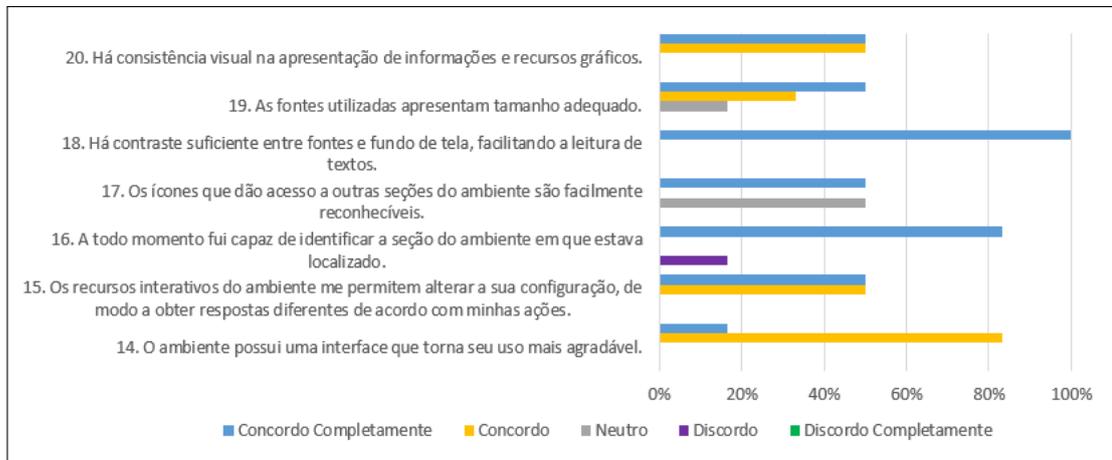


Fonte – o autor.

Na avaliação de interface, como a Figura 48 apresenta, 83% dos estudantes afirmaram que concordam que o ambiente possui uma interface que torna seu uso mais agradável e 17% disseram concordar completamente. 83% dos alunos concordam completamente que foram capazes de identificar a seção do ambiente em que estavam localizados. 50% também concordaram completamente que os ícones que dão acesso a outras seções do ambiente são facilmente reconhecíveis, que as fontes utilizadas apresentam tamanho adequado e que há consistência visual na apresentação de informações e recursos gráficos.

Para a avaliação da prática de programação, 80% dos estudantes concordaram completamente que a atividade prática de programação lhes ajudou a entender e conhecer melhor a arquitetura do *framework Context Toolkit* proposta por Dey e Abowd. 60% dos alunos também concordaram completamente que essa atividade lhes ajudou a entender e conhecer melhor o fluxo de informações em aplicações sensíveis ao contexto baseadas na arquitetura *Context Toolkit*. Segundo os alunos, o mais interessante dessa prática de programação foram “o nível de organização do projeto e o uso de padrões de projeto”, “o código bem estruturado”, “o código muito bem escrito, fácil de ler e de entender a arquitetura” e “a experiência de estender o *framework*”. No item que pedia para descrever o que os alunos mudariam nesse exercício, a maioria respondeu que não mudaria nada e outro disse que as perguntas que exigem respostas discursivas feitas no questionário do Apêndice F dessa prática de programação tomam tempo

Figura 48 – Respostas da avaliação de interface dos alunos.



Fonte – o autor.

demais da atividade.

6.1.5 Análise e discussões

Pelas respostas, os estudantes avaliaram de forma bastante positiva a usabilidade e a interface do ambiente. Além disso, a proposta das atividades foi bem aceita e avaliada pelos alunos, que executaram com êxito o que foi pedido durante as atividades de simulação e a prática de programação.

Um ponto limitante na avaliação do Lucy é o número de usuários que participaram da avaliação. Não foi possível ainda mensurar de fato se houve melhoria no rendimento e na participação deles nas aulas de Computação Ubíqua a partir da utilização do ambiente, pois há necessidade de mais tempo de observação, avaliação e análise com a utilização do Lucy em sala de aula e em diferentes turmas. No entanto, foi possível observar uma melhor definição das respostas dos alunos após a realização da simulação, conforme apresenta o Quadro 6.

6.2 Avaliação com professores

Esta seção trata de descrever o processo e os resultados da avaliação do ambiente Lucy realizado com professores ligados à área de Computação Ubíqua. Ao todo, 6 professores de diferentes universidades brasileiras testaram o Lucy e o avaliaram considerando sua utilidade, aplicabilidade, usabilidade e interface.

6.2.1 Perfil dos professores

O grupo avaliado era composto por 6 professores, sendo cinco do sexo masculino e um do sexo feminino. Todos os professores possuíam pesquisas ligadas à Computação Ubíqua e alguns ainda ministravam disciplinas relacionadas ao tema, tendo todos bastante experiência com o assunto tratado. Do total, havia dois professores da UFC, um da UFG, um da UFPE, um da Universidade Federal Rural de Pernambuco (UFRPE) e um da Universidade Federal de Uberlândia (UFU).

6.2.2 Materiais e métodos

As avaliações realizadas com os professores foram de forma presencial com os dois da UFC e à distância por Skype com os demais, pois estes eram de outros estados brasileiros, o que inviabilizava o teste de forma presencial. Em ambos os casos, os professores acessaram o Lucy, realizaram algumas configurações na simulação de Sensibilidade ao Contexto e observaram os resultados daquilo que fizeram no *smartphone*.

Todos os professores pesquisados foram esclarecidos da utilização do Lucy em aulas práticas de Computação Ubíqua e da avaliação do ambiente feita com os alunos. O questionário do Apêndice E, já aplicado com os estudantes, foi também adotado para a avaliação dos professores, a fim de se verificar aspectos pedagógicos, de usabilidade e de interface do ambiente. Além desse, um outro questionário (Apêndice H) com perguntas discursivas foi usado para mensurar a satisfação dos professores com a proposta deste trabalho.

6.2.3 Procedimentos

As avaliações com os professores foram feitas de forma individual e duraram em média 40 minutos. Para estes testes, os professores, primeiramente, assistiram a um vídeo que apresenta o Lucy e seu funcionamento. Depois foi descrito para eles como ocorreu a utilização do Lucy em sala de aula, como foram as atividades desenvolvidas com ele e como se deu sua avaliação com os alunos.

Em seguida, os professores acessaram o ambiente e realizaram duas ações guiadas na simulação, observando o resultado do que faziam no *smartphone*. As instruções dadas aos professores durante a execução da simulação no Lucy foram: “habilite o sensor de luminosidade e seu *Widget* correspondente” e “Na parte de tomada de decisão, habilite um *trigger* simples e

defina => quando a luminosidade for igual a (algum valor escolhido pelo professor), a lanterna será ligada.” Após o passo a passo, os professores ficaram livres para fazer outras ações que quisessem dentro do ambiente.

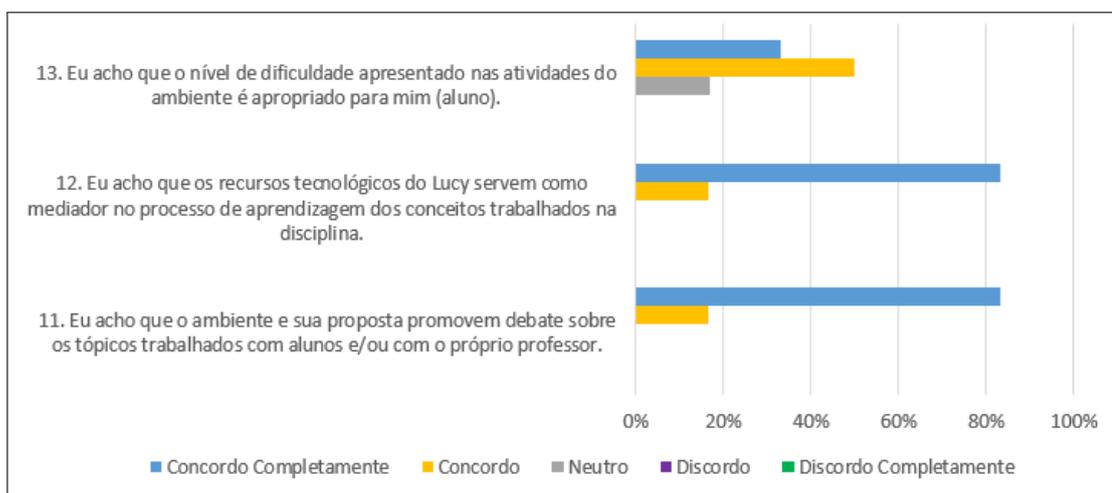
Por último, os pesquisados responderam a dois questionários. O primeiro era relativo a informações pedagógicas, de usabilidade e de interface (Apêndice E). O segundo (Apêndice H) estava relacionado à satisfação do professor com o Lucy. Os resultados obtidos com estes testes são apresentados na próxima seção.

6.2.4 Resultados

Com relação à usabilidade do ambiente, os professores classificaram-na como muito satisfatória, com a pontuação de 84,3 pontos obtidos conforme as regras do SUS. Com essa avaliação dos professores, significa dizer que o Lucy está na escala “A” de classificação do SUS.

Nos resultados mostrados pela Figura 49 referente à avaliação de aspectos pedagógicos, 83% dos professores responderam que concordam completamente que o ambiente e sua proposta promovem debate sobre os tópicos trabalhados com alunos e/ou com o próprio professor e ainda que os recursos tecnológicos do Lucy servem como mediador no processo de aprendizagem dos conceitos trabalhados na disciplina. 50% dos professores também concordam que o nível de dificuldade apresentado nas atividades do ambiente é apropriado para os alunos.

Figura 49 – Respostas da avaliação de aspectos pedagógicos dos professores.



Fonte – o autor.

A Figura 50 apresenta os resultados da avaliação de interface, na qual 67% dos professores concordam que o ambiente possui uma interface que torna seu uso mais agradável.

50% dos professores concordam que os recursos interativos do ambiente permitem alterar sua configuração, de modo a obter respostas diferentes de acordo com as ações. 50% concordam também que a todo momento foram capazes de identificar a seção do ambiente em que estavam localizados e que as fontes utilizadas apresentam tamanho adequado. 67% dos professores estão de acordo que os ícones que dão acesso a outras seções do ambiente são facilmente reconhecíveis, que há contraste suficiente entre fontes e fundo de tela, facilitando a leitura de textos e também que há consistência visual na apresentação de informações e recursos gráficos.

Figura 50 – Respostas da avaliação de interface dos professores.



Fonte – o autor.

Na área de sugestão, alguns professores relataram que o ambiente é essencial ao ensino de conceitos ubíquos, pois existem poucos materiais para esse fim, além da proposta ser simples de aplicar em sala de aula. Outros relataram que não se sentiram a vontade com as cores utilizadas na ferramenta Web e que as configurações deveriam afetar imediatamente a aplicação móvel, sem precisar esperar pela mudança de estado dos sensores correspondentes.

Os resultados obtidos por meio do questionário de satisfação (Apêndice H) mostram que todos os professores pesquisados consideraram as atividades propostas pelo Lucy aceitáveis para os alunos que estudam conceitos de Computação Ubíqua. Os pesquisados afirmaram que a proposta do Lucy seria adequada tanto para graduandos como para pós-graduandos. Quando perguntados sobre quais eram suas impressões a respeito do ambiente, os professores falaram que consideram a proposta bastante interessante, que deve ser compartilhada com outras pessoas e que é um ponto de partida para a exploração de outros conceitos. Eles afirmaram ainda não conhecer outros recursos com proposta similar ao do Lucy.

Segundo os professores, alguns aspectos no ambiente ainda podem ser melhorados,

tais como: “indicar através de alertas se existem *triggers* conflitantes”, “usar mais sensores e opções de configuração” e “disponibilizar edição e ajustes de conteúdo para que os professores estruturarem suas aulas de forma mais particular às suas necessidades”. Todos os pesquisados concordam com a forma de utilização do Lucy em sala de aula, não tendo alterações ou objeções a fazer na abordagem realizada com o ambiente nas aulas práticas relacionadas à Sensibilidade ao Contexto.

6.2.5 Análise e discussões

Os professores ficaram satisfeitos com o ambiente e avaliaram sua usabilidade e interface de forma bastante positiva. Além disso, eles concordaram que o nível das atividades está apropriado para seus alunos e que o Lucy estimula o debate dos conceitos estudados em sala de aula. Um professor afirmou inclusive que já conseguia ver o Lucy sendo usado em suas aulas de Sensibilidade ao Contexto e que gostaria de começar a usá-lo de fato com seus alunos.

A amostra de professores avaliados ainda é pequena. Faz-se necessário a realização de testes com um maior número de participantes e que estes utilizem também o ambiente em suas aulas a fim de testar ainda mais as possibilidades e a efetividade desta proposta. Apesar disso, os resultados conseguidos apontam a necessidade e a aceitação do Lucy como recurso de simples aquisição e utilização para a prática de conceitos de Computação Ubíqua.

6.3 Conclusão

Neste capítulo, foram relatadas as diversas avaliações realizadas com professores e alunos para qualificar o Lucy. Com isso, os resultados obtidos foram discutidos também neste capítulo e se mostraram bastante satisfatórios no que diz respeito a boa aceitação das atividades práticas propostas e dos recursos (Web e móvel) utilizados.

Estudantes e professores deram um retorno bastante positivo sobre o ambiente de modo geral, seu funcionamento, sua interface e sua usabilidade. Alguns avaliados sugeriram também ajustes e aperfeiçoamentos a fim de refinar ainda mais a proposta. Algumas limitações das avaliações também foram discutidas, o que ainda impossibilita uma análise mais precisa e abrangente para os resultados obtidos. O próximo capítulo traz a conclusão desta dissertação, mostrando os resultados obtidos, as limitações e trabalhos futuros relacionados a este tema.

7 CONCLUSÃO

Nesta dissertação de mestrado, foi apresentado o ambiente Lucy, o conceito de Sensibilidade ao Contexto desenvolvido e implementado como tópico do ambiente e as suas respectivas atividades práticas. Neste capítulo, estão sumarizados os principais resultados, contribuições e limitações do trabalho desenvolvido nesta dissertação. Também são apresentadas as propostas de trabalhos futuros que darão continuidade a esta pesquisa.

Na Seção 7.1, os resultados alcançados são listados. A Seção 7.2 indica as principais limitações da pesquisa. A Seção 7.3 apresenta a produção bibliográfica da autora deste trabalho ao longo do mestrado. Por fim, a Seção 7.4 traz propostas de trabalhos futuros derivados deste trabalho.

7.1 Resultados alcançados

Este trabalho apresentou a proposta de um ambiente voltado para o ensino prático de conceitos ubíquos, a ser utilizado na graduação e pós-graduação. Esta pesquisa de mestrado busca fechar uma lacuna existente e satisfazer as necessidades de professores e alunos de por em prática conceitos ubíquos estudados com o auxílio de recursos encontrados de forma simples e gratuita, como uma página Web e uma aplicação móvel, por exemplo.

Como mencionado no decorrer desta dissertação, os materiais disponíveis para auxiliar as práticas de conceitos ubíquos são bastante escassos, complexos ou foram feitos inicialmente para outras finalidades. Os professores e alunos pesquisados relataram as dificuldades da realização de aulas práticas em disciplinas relacionadas à Computação Ubíqua, ressaltando a existência de conceitos muito abstratos, a grande carga de conteúdo para pouco tempo de aula, a dificuldade de executar aulas teóricas e práticas de forma coordenadas e a falta de ferramentas e materiais para auxiliar as atividades práticas. Nesse sentido, duas questões de pesquisa nortearam o desenvolvimento deste trabalho.

Questão 1 - (QP1) - Como se dá o ensino de Computação Ubíqua nas universidades?

Essa questão de pesquisa foi respondida por meio de estudos do estado da arte relacionados ao ensino e à aprendizagem de Computação Ubíqua, além de investigações feitas com alunos e professores de diferentes instituições de ensino superior no Brasil e fora dele. Essas pesquisas foram apresentadas nos Capítulos 2 e 3 respectivamente e teve como objetivo conhecer e mostrar o cenário de alguns projetos e algumas disciplinas de graduação e de pós-graduação

relacionados à Computação Ubíqua, sendo isto também uma das contribuições desta pesquisa. É importante salientar que as informações colhidas nesta etapa foram levadas em consideração para o desenvolvimento e aperfeiçoamento do ambiente Lucy.

Questão 2 - (QP2) - Quais características devem existir em um ambiente de apoio ao ensino de Computação Ubíqua de forma a favorecer a compreensão dos estudantes acerca de conceitos teóricos dessa área?

Para responder a essa questão de pesquisa, foi usado como base os estudos dos trabalhos científicos que descrevem materiais e métodos para as aulas práticas de Computação Ubíqua. Além disso, também foi levado em consideração as respostas de professores e alunos pesquisados. Com isso, chegou-se a proposta de um ambiente integrado por diversos conceitos ubíquos que possuem seções de teoria, simulação e prática. Este ambiente, o Lucy, foi então desenhado e desenvolvido como a principal proposta desta dissertação, que é apresentada e descrita no Capítulo 4.

A prova de conceito descrita no Capítulo 5 apresenta o tema de Sensibilidade ao Contexto como componente constituinte do Lucy e ilustra ainda a divisão deste tópico em seções de teoria, simulação e prática. Esse tema representa um exemplo de como podem ser integrados ao ambiente diversos outros conceitos de Computação Ubíqua e o que deve existir dentro de cada seção relativa a esses conceitos criados para a promoção de aulas práticas.

Diferentes tipos de avaliação do Lucy foram feitas com alunos e professores para dimensionar a qualidade do ambiente proposto. Os resultados e discussões estão expostos no Capítulo 6. Com base nesses resultados, é possível afirmar que a aceitação do ambiente pelos usuários foi bastante positiva, sendo a usabilidade, a interface e a utilidade do Lucy muito bem estimadas pelos seus avaliadores.

Comparando o Lucy com os trabalhos relacionados apresentados no Capítulo 2, observa-se como ele se difere das demais ferramentas discutidas e como ele contribui na execução de aulas práticas relacionadas à Computação Ubíqua. O Quadro 7 expõe as características do Lucy em relação as dos demais trabalhos estudados. O ambiente proposto se destaca principalmente pelo uso de recursos comumente presentes no cotidiano da maioria dos estudantes de cursos ligados à tecnologia. O Quadro 8 mostra a atuação do Lucy em aulas práticas de Computação Ubíqua em comparação com demais trabalhos. A utilização do Lucy em sala de aula se difere também pela concentração das atividades em distintas seções, que se dividem em teoria, simulação e prática dentro do mesmo ambiente.

Quadro 7 – Comparativo entre os materiais apresentados para as aulas práticas incluindo o Lucy.

	Trabalha diferentes conceitos de Computação Ubíqua	Nível de Conhecimento de programação exigido	Vantagens	Desvantagens
SenseBoard (RICHARDS <i>et al.</i> , 2012)	Sim	Sem exigências	<ul style="list-style-type: none"> – Permite trabalhar diversos conceitos de Computação Ubíqua. – Não exige conhecimentos prévios em programação. 	<ul style="list-style-type: none"> – O hardware SenseBoard não está mais disponível. – A linguagem em blocos Sense está instável.
LEGO (SILVICIVIDJIAN, 2015)	Sim	Sem exigências	<ul style="list-style-type: none"> – Permite trabalhar diversos conceitos de Computação Ubíqua. – Não exige conhecimentos prévios em programação. 	<ul style="list-style-type: none"> – Custo médio/alto para aquisição e manutenção. – Limitação dos tipos de sensores e atuadores utilizados.
Phidgets (CHALMERS, 2015)	Sim	Conhecimentos de programação	<ul style="list-style-type: none"> – Permite trabalhar diversos conceitos de Computação Ubíqua. – Fornece APIs em várias linguagens de programação. 	<ul style="list-style-type: none"> – Custo médio/alto para aquisição. – Não há um ambiente/sistema que relacione aulas teóricas e práticas.
Simulador de sistemas sensíveis ao contexto (MARTIN; NURMI, 2006)	Não – Apenas sensibilidade ao contexto	Conhecimentos de programação em Java	<ul style="list-style-type: none"> – Permite a visualização ilustrada do conceito estudado. – Possibilita ao aluno a criação própria de uma simulação. 	<ul style="list-style-type: none"> – Não foi criado para o ensino-aprendizagem de conceitos ubíquos. – Não há interação com os agentes na interface de simulação criada.
Lucy	Aborda apenas Sensibilidade ao Contexto no momento	Simulação: não exige conhecimentos de programação. Prática de programação: exige conhecimentos de programação em Java.	<ul style="list-style-type: none"> – Concentra seções relacionadas a atividades teóricas e práticas em um mesmo ambiente. – Utiliza recursos comuns do cotidiano de estudantes da área tecnológica. – Foi desenvolvido especialmente para o ensino e a aprendizagem de conceitos de Computação Ubíqua. 	Embora seja um ambiente voltado para o ensino de diversos conceitos de Computação Ubíqua, apenas o conceito de Sensibilidade ao Contexto foi implementado até o final desta dissertação de mestrado.

Fonte – o autor.

Quadro 8 – Comparativo entre as propostas de cursos relacionados à Computação Ubíqua incluindo o Lucy.

	Local	Público-Alvo	Divisão das aulas/atividades	Recursos usados	Estrutura das aulas práticas
Curso de Computação Ubíqua (CHALMERS, 2015)	Universidade de Sussex, no Reino Unido	Estudantes de graduação e pós-graduação	Exposição teórica, seminários e práticas em laboratório	Phidgets	Desenvolvimento de experimentos direcionados usando Phidgets.
Curso de Computação Ubíqua a distância (GOUMOPOULOS <i>et al.</i> , 2017)	Hellenic Open University, na Grécia	Estudantes de pós-graduação a distância	Atividades teóricas e práticas	Android, LEGO Mindstorms, Phidgets, Arduino, 123d circuits, Wireshark e laboratório remoto de Arduino	Desenvolvimento de aplicações direcionadas com as tecnologias citadas e resolução de exercícios.
Curso de Inteligência Ambiental (CORNO <i>et al.</i> , 2016)	Politecnico di Torino, na Itália	Estudantes de graduação	Atividades teóricas e práticas	Raspberry Pi, Android, protocolos Web e sistemas <i>smart home</i>	Atividades práticas e exercícios em sala de aula, exercícios guiados no laboratório e trabalhos em grupo no laboratório.
Sensibilidade ao Contexto com Lucy	Universidade Federal do Ceará, no Brasil	Estudantes de graduação e pós-graduação	Atividades teóricas e práticas (utilizando as seções de simulação e prática de programação para Android)	Lucy (página Web e smartphone Android)	Atividades práticas com a simulação e a programação de código Android.

Fonte – o autor.

7.2 Limitações

Devido às limitações de duração do tempo de uma pesquisa de mestrado, o ambiente Lucy possui apenas o conceito de Sensibilidade ao Contexto implementado até o final desta dissertação. No entanto, objetiva-se desenvolver outros conceitos nos mesmos moldes do já existente, com seções de teoria, simulação e prática. Esses novos conceitos podem seguir as indicações dos professores e alunos ouvidos no *survey*, que apontaram *middleware*, adaptabilidade, computação móvel e heterogeneidade como conceitos importantes de Computação Ubíqua.

Outra limitação é o fato do ambiente ainda não ter telas para edição dos conceitos e de seus conteúdos. Alguns professores avaliados falaram da necessidade dessa função, pois gostariam de disponibilizar apenas alguns conceitos, escolher o que colocar na seção de teoria, acrescentar ou reduzir os sensores utilizados pela simulação ou ainda modificar as regras desta e alterar a atividade da prática de programação.

Vale ainda ressaltar que as avaliações realizadas com o Lucy necessitam ser feitas

com um número maior de usuários para que seja possível a maior generalização dos resultados observados. Para que se possa também mensurar o impacto do ambiente na aprendizagem dos alunos, é necessária sua utilização e avaliação em diferentes turmas ao longo de vários períodos.

7.3 Produção bibliográfica

Durante o período de desenvolvimento desta pesquisa de mestrado, quatro submissões foram realizadas para eventos nacionais e internacionais. Duas delas foram publicadas:

- *Uma Proposta de Ferramenta de Apoio ao Ensino Prático de Computação Ubíqua*. Publicado no XVI Workshop de Teses e Dissertações (WTD) do XXII Simpósio Brasileiro de Sistemas Multimídia e Web (WebMedia 2016), apresentando a concepção e o desenvolvimento inicial de uma ferramenta voltada para auxiliar as aulas práticas de Computação Ubíqua. Autores: Maria Joelma Pereira Peixoto, Windson Viana de Carvalho e Wellington Wagner Ferreira Sarmiento.
- *Ferramenta de Apoio ao Ensino Prático de Computação Ubíqua*. Publicado como *full-paper* no XXI Congresso Internacional de Informática Educativa (TISE 2016), apresentando uma ferramenta de apoio para as aulas práticas de Computação Ubíqua com suas primeiras avaliações com alunos de uma disciplina relacionada à Computação Ubíqua. Foram também apresentados os primeiros resultados e impressões obtidos. Autores: Maria Joelma Pereira Peixoto, Wellington Wagner Ferreira Sarmiento e Windson Viana de Carvalho.
- *Lucy: An Environment for Practical Ubiquitous Computing Classes*. Aceito inicialmente como *full paper abstract* no Frontiers in Education (FIE 2017), não foi publicado por falta de recursos financeiros. O trabalho está sendo adaptado e teve seu *abstract* submetido para a categoria de artigos completos no Congresso Brasileiro de Informática da Educação (CBIE 2017) e, caso seja aceito, terá seu texto completo submetido para publicação. Esse trabalho apresenta um ambiente para o ensino e a aprendizagem de conceitos de Computação Ubíqua, tendo sido avaliado com vários alunos de diferentes disciplinas relacionadas à Computação Ubíqua. Autores: Maria Joelma Peixoto, Pedro Pinto, Fernando Trinta, Wellington Ferreira Sarmiento e Windson Viana.

7.4 Trabalhos futuros

Esta dissertação de mestrado é o ponto de partida para o desenvolvimento de um completo ambiente especialmente criado para o ensino e a aprendizagem de Computação Ubíqua. Pelo fato do Lucy não está totalmente finalizado, os trabalhos futuros desta pesquisa podem ser divididos em categorias de curto e longo prazo.

O primeiro trabalho futuro de curto prazo se refere a encontrar professores e disciplinas ligados à Computação Ubíqua que possam usar e avaliar o ambiente em suas aulas. Assim, será possível obter mais dados relacionados à usabilidade, à interface, aos aspectos pedagógicos e à satisfação com o ambiente e suas atividades propostas.

Outro trabalho futuro de curto prazo diz respeito ao desenvolvimento de outros conceitos relacionados à Computação Ubíqua que utilizem a mesma estrutura - teoria, simulação e prática - e as mesmas tecnologias anteriormente usadas, páginas Web e *smartphones*. Os próximos conceitos devem seguir a mesma estrutura de organização e de uso à semelhança do que foi feito com o conceito de Sensibilidade ao Contexto criado dentro do ambiente Lucy.

Como um trabalho de longo prazo, tem-se o desenvolvimento das seções de edição destinadas aos professores. Com isso, será possível que o professor configure o ambiente, de maneira mais simplificada, de acordo com suas necessidades e a realidade das aulas práticas de cada curso de Computação Ubíqua. Deverá ser possível editar os conteúdos de todas as seções de cada conceito, assim como escolher os sensores que se quer usar na simulação e as regras que serão definidas pelos alunos ao longo da realização das atividades com a simulação.

Outro trabalho futuro de longo prazo é a execução de avaliação pedagógica constante durante o uso do Lucy no decorrer de vários semestres e várias disciplinas para que se possa mensurar o impacto positivo gerado na aprendizagem de estudantes de Computação Ubíqua. Assim será possível identificar a real utilidade do ambiente, destacando-se os pontos de mudanças positivas desde o início de sua utilização.

REFERÊNCIAS

- ABNT. **Ergonomia da interação humano-sistema. Parte 210: Projeto centrado no ser humano para sistemas interativos.** Rio de Janeiro, 2011.
- ARAÚJO, R. B. **Computação ubíqua: princípios, tecnologias e desafios.** 2003. XXI Simpósio Brasileiro de Redes de Computadores.
- BARESI, L.; NITTO, E. D.; GHEZZI, C. Towards open-world software: Issues and challenges. **Computer, IEEE Computer Society**, v. 39, p. p. 36–43, 2006.
- BOYCE, C.; NEALE, P. **CONDUCTING IN-DEPTH INTERVIEWS: A Guide for Designing and Conducting In-Depth Interviews for Evaluation Input.** 2006. PATHFINDER INTERNATIONAL: CONDUCTING IN-DEPTH INTERVIEWS.
- BROOKE, J. Sus: A retrospective. **Journal of Usability Studies**, v. 8, p. 29–40, 2013.
- CACHO, C. E. A.; SOUZA, P. S. L.; BRUSCHI, S. M.; BARBOSA, E. F.; TIOSSO, F. An interactive approach for the teaching of virtual memory using open educational resources. **SAC '16 Proceedings of the 31st Annual ACM Symposium on Applied Computing**, 2016.
- CARVALHO, W.; ANDRADE, R.; MARCAL, E.; MAIA, M.; LIMA, J. V. de; GENSEL, J.; JUNIOR, J. C. F.; SÁNCHEZ, J. Towards context-aware and mobile e-learning applications. **Nuevas Ideas en Informática Educativa, TISE**, 2011.
- CHALMERS, D. Pervasive computing as a classroom-based course. **Pervasive computing - IEEE**, p. pp. 70–78, 2015.
- COOMANS, S.; LACERDA, G. S. Petese, a pedagogical ergonomic tool for educational software evaluation. **6th International Conference on Applied Human Factors and Ergonomics**, 2015.
- CORNO, F.; RUSSIS, L. D.; BONINO, D. Educating internet of things professionals: The ambient intelligence course. **IT Professional**, 2016.
- COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T.; BLAIR, G. **Distributed Systems: Concepts and Design.** 5th.. ed. [S.l.]: Addison Wesley, 2011.
- COUTAZ, J.; CROWLEY, J. L.; DOBSON, S.; GARLAN, D. Context is key. **Communications of the ACM**, 2005.
- DEY, A. K. Understanding and using context. **Personal and Ubiquitous Computing**, 2001.
- DEY, A. K.; ABOWD, G. D. The context toolkit: Aiding the development of context-aware applications. **Workshop on Software Engineering for Wearable and Pervasive Computing**, 1999.
- DUARTE, P. A. **Uma Abordagem Dirigida por Modelos para a Configuração de Aquisição de Contexto Intermediada por Middleware.** Dissertação (Dissertação de Mestrado) — Universidade Federal do Ceará, Fortaleza, Ceará, Brasil., 2014.
- DUARTE, P. A.; SILVA, L. F. M.; GOMES, F. A.; VIANA, W.; TRINTA, F. M. Dynamic deployment for context-aware multimedia environments. **WebMedia '15 Proceedings of the 21st Brazilian Symposium on Multimedia and the Web**, 2015.

FERREIRA, D.; KOSTAKOS, V.; DEY, A. K. Aware: mobile context instrumentation framework. **Frontiers in ICT**, Frontiers, v. 2, p. 6, 2015.

FILHO, C. A. B. de Q. **Um Mecanismo de Tratamento de Exceções Sensível ao Contexto para Sistemas Ubíquos Orientados a Tarefas**. Dissertação (Dissertação de Mestrado) — Universidade Federal do Ceará, Fortaleza, Ceará, Brasil., 2012.

GOUMOPOULOS, C.; NICOPOLITIDIS, P.; GAVALAS, D.; KAMEAS, A. A distance learning curriculum on pervasive computing. **International Journal of Continuing Engineering Education and Life Long Learning**, 2017.

HAMDAOUI, B.; ALSHAMMARI, T.; GUIZANI, M. Exploiting 4g mobile user cooperation for energy conservation: Challenges and opportunities. **IEEE Wireless Communications**, 2013.

ISO. **Systems and software engineering — Vocabulary**. 2011.

KNAPPMAYER, M.; KIANI, S. L.; REETZ, E.; BAKER, N.; TONJES, R. Survey of context provisioning middleware. **IEEE**, 2013.

KRUMM, J. **Ubiquitous Computing Fundamentals**. Redmond, Washington. [S.l.]: U.S.A.: CRC Press, 2010.

LIMA, F. F. **SysSU - Um Sistema de Suporte para Computação Ubíqua**. Dissertação (Dissertação de Mestrado) — Universidade Federal do Ceará, Fortaleza, Ceará, Brasil, 2011.

MARTIN, M.; NURMI, P. A generic large scale simulator for ubiquitous computing. **Mobile and Ubiquitous Systems: Networking & Services**, 2006.

MELNIKOV, A.; FETTE, I. **The WebSocket Protocol**. RFC Editor, 2011. RFC 6455. (Request for Comments, 6455). Disponível em: <<https://rfc-editor.org/rfc/rfc6455.txt>>.

NIEUWDORP, E. The pervasive discourse: an analysis. **Computers in Entertainment (CIE) - Interactive TV**, 2007.

OLIVEIRA, F. S. de; SANTOS, S. Pblmaestro: A virtual learning environment for the implementation of problem-based learning approach in computer education. **Frontiers in Education Conference (FIE), 2016 IEEE**, 2016.

QUINTINO, J. P.; FERRAZ, C. A. G. A middleware for context-aware integration of power system software. **EATIS '14 Proceedings of the 7th Euro American Conference on Telematics and Information Systems**, 2014.

REATEQUI, E.; BOFF, E.; FINCO, M. D. Proposta de diretrizes para avaliação de objetos de aprendizagem considerando aspectos pedagógicos e técnicos. **CINTED-UFRGS. Novas Tecnologias na Educação**, 2010.

RICHARDS, M.; PETRE, M.; BANDARA, A. Starting with ubicomp: Using the senseboard to introduce computing. **SIGCSE'12**, p. pp. 583–588, 2012.

SANTOS, E. O. Ambientes virtuais de aprendizagem: por autorias livre, plurais e gratuitas. **Revista FAEBA**, v. 12, n. 18, 2003.

SARMENTO, W. W. F.; PAULA, P. S.; FILHO, P. T. C. P.; PALLARD, G. A. L.; SOUZA, J. N.; BARROSO, G. C.; PEQUENO, M. C. u-lab: A ubiquitous computing based architecture to labs works using wireless sensor network and radio-frequency identification. In: **Proceedings of the 6th Euro American Conference on Telematics and Information Systems**. New York, NY, USA: ACM, 2012. (EATIS '12), p. 260–266. ISBN 978-1-4503-1012-3. Disponível em: <<http://doi.acm.org/10.1145/2261605.2261644>>.

SATYANARAYANAN, M. Pervasive computing: vision and challenges. **IEEE Personal Communications, IEEE Communications Society**, p. p. 10–17, 2001.

SAURO, J. **Measuring Usability with the System Usability Scale (SUS)**. 2011. Disponível em: <<https://measuringu.com/sus/>>.

SILVIS-CIVIDJIAN, N. Teaching pervasive computing to cs freshmen: A multidisciplinary approach. **ITiCSE '15**, p. pp. 195–200, 2015.

TONG, A.; SAINSBURY, P.; CRAIG, J. Consolidated criteria for reporting qualitative research (coreq): a 32-item checklist for interviews and focus groups. **International Journal for Quality in Health Care**, 2007.

VEIGA, E. F.; MARANHÃO, G. M. e; NETO, R. de F. B. Apoio ao desenvolvimento de aplicações de tempo real sensíveis a contexto semântico. **20th Brazilian Symposium on Multimedia and the Web**, 2014.

VIANA, W. **Mobility and Context-awareness for Personal Multimedia Management: CoMMedia**. Tese (Doutorado) — Université Joseph-Fourier - Grenoble I, fev. 2010.

WEISER, M. The computer for the 21st century. **Scientific American**, v. 265, n. 3, p. 66–75, set. 1991.

WENDT, N.; JULIEN, C. Mason: An open development contextual sensing framework enabling reactive applications. In: **Proceedings of the International Conference on Mobile Software Engineering and Systems**. New York, NY, USA: ACM, 2016. (MOBILESoft '16), p. 100–110. ISBN 978-1-4503-4178-3. Disponível em: <<http://doi.acm.org/10.1145/2897073.2897099>>.

WOOD, S.; ROMERO, P. User-centred design for a mobile learning application. **Proceedings of the 3rd Mexican Workshop on Human Computer Interaction**, 2010.

WOODSON, W. E.; TILMAN, B.; TILMAN, P. **Human factors design handbook**. 1981.

APÊNDICE A – PESQUISA COM PROFESSORES SOBRE AULAS DE COMPUTAÇÃO UBÍQUA

Questionário com a finalidade de conhecer o cenário de ensino de Computação Ubíqua e verificar a necessidade de um ambiente para o ensino prático de conceitos ubíquos.

* Required

1. Nome *

2. Universidade *

3. 1) Há quanto tempo você é professor? *

4. 2) Qual(is) o(s) nome(s) da(s) disciplina(s) que você ministra e que possui(em) conceitos de Computação Ubíqua? *

5. 3) Você já pesquisava sobre Computação Ubíqua antes de ministrar sua(s) disciplina(s)? *

6. 4) A(s) disciplina(s) que você ministra e que possui(em) conceitos de Computação Ubíqua é(são) da Graduação ou da Pós-graduação? *

7. 5) Há quanto tempo você ministra a(s) disciplina(s) acima mencionada(s)? *

8. 6) Qual a quantidade de aulas consagradas ao tema de Computação Ubíqua em sua(s) disciplina(s)? *

9. 7) Quantas delas são aulas práticas? *

10. 8) Você utiliza algum ambiente/kit para ensinar Computação Ubíqua? Se sim, qual? *

11. 9) Você poderia citar ao menos quatro conceitos de Computação Ubíqua que você considera importante e que consagra mais de uma aula para cada um deles? (ex.: sensibilidade ao contexto) *

12. 10) Quais as dificuldades que você encontra no ensino desses conceitos? *

13. 11) O que você acha da ideia de haver um ambiente interativo(OA, simulador, ferramenta ligada a um middleware) para o ensino de Computação Ubíqua? *

14. 12) Na sua opinião, o que esse ambiente interativo deveria ter? *

15. 13) Você veria como uma prática interessante a programação, por parte do aluno, de elementos de um sistema ubíquo? Se sim, qual plataforma de programação você vê como ideal? *

16. 14) Qual o conceito dos quatro elencados por você acima se caracteriza como o mais importante para o ambiente interativo? *

APÊNDICE B – LEVANTAMENTO DE REQUISITOS PARA UM AMBIENTE INTERATIVO DE ENSINO DE COMPUTAÇÃO UBÍQUA - ALUNO

* Required

1. 1) Qual sua instituição de ensino superior? *

2. 2) Qual seu curso? *

3. 3) Em que ano você cursou alguma disciplina que abordasse mais profundamente os conceitos de Computação Ubíqua? *

Mark only one oval.

- 2013
- 2014
- 2015
- 2016.1
- Estou fazendo (2016.2)
- Other: _____

4. 4) Você fez(faz) essa disciplina: *

Mark only one oval.

- Na graduação
- Na pós-graduação

5. 5) Você tinha/tem grande conhecimento/experiência em programação no momento da realização da disciplina? *

Mark only one oval.

- Concordo totalmente
- Concordo
- Não concordo nem discordo
- Discordo
- Discordo totalmente

6. 6) Você teve(tem) aulas práticas de Computação Ubíqua? *

Mark only one oval.

- Concordo totalmente
- Concordo
- Não concordo nem discordo
- Discordo
- Discordo totalmente

7. 7) Qual ambiente de desenvolvimento ou kit você utilizou(utiliza) nas aulas práticas?
(Caso tenha tido aulas práticas)

8. 8) Quais conceitos foram mais difíceis de entender na ausência de aulas práticas? *

Check all that apply.

- Heterogeneidade
- Adaptação
- Middleware
- Sensibilidade ao Contexto
- Sistemas Baseados em Localização
- Other: _____

9. 9) Você gostaria que tivesse um ambiente voltado especificamente para a prática dos conceitos de Computação Ubíqua? *

Mark only one oval.

- Concordo totalmente
- Concordo
- Não concordo nem discordo
- Discordo
- Discordo totalmente

10. 10) O que você acha que deveria ter nesse ambiente? *

Check all that apply.

- Explicação teórica
- Representação gráfica dos conceitos teóricos
- Simulação do comportamento de sensores e serviços a partir da inserção de parâmetros
- Programação de códigos Android, pré-moldados pelo ambiente, para visualização, em dispositivos móveis, dos conceitos ubíquos.
- Other: _____

11. 11) Escreva aqui suas sugestões ou observações sobre aulas práticas no ensino de Computação Ubíqua.

APÊNDICE C – QUESTIONÁRIO PRÉ E PÓS PRÁTICA

* Required

1. **1) Em que momento você está respondendo este questionário? ***

Mark only one oval.

- Antes da prática
 Depois da prática

2. **2) Coloque um identificador de sua escolha. (Este mesmo identificador deve ser usado nos dois momentos de resposta deste questionário) ***

3. **3) Como você pode diferenciar os widgets de interpretadores? ***

4. **4) O que você entende como papel ou função de um Agregador de contexto? E do conceito de elemento contextual de alto nível? ***

5. **5) O que caracteriza os triggers? ***

APÊNDICE D – QUESTIONÁRIO GUIA DA ATIVIDADE DE SIMULAÇÃO

Aula Prática de Arquiteturas de Computação Ubíqua

Número de Matrícula:

Preparatório

- 1- Abra o ambiente Lucy, crie uma conta e acesse a área de simulação do Context Toolkit. Instale também o aplicativo móvel em seu aparelho Android. Ele está configurado para se comunicar com o ambiente Web conforme o código mostrado na tela de simulação.
- 2- A ferramenta Web funciona com o princípio de habilitação dos componentes do framework Context Toolkit: sensores, widgets, interpretadores, agregadores e também acrescenta uma etapa de tomada de decisão.

Aquecimento

- 1- Habilite o sensor de GPS na ferramenta e observe a aplicação móvel. Neste ponto, o sensor foi habilitado. Com a habilitação do widget, uma primeira informação de baixo nível é produzida
- 2- Habilite o interpretador de localização e veja o que acontece. Neste ponto, temos um *gazeteer* que transforma coordenadas GPS em endereços postais.

Prática

Atenção: Para responder a prática, inclua imagens das configurações da ferramenta Web e do aplicativo móvel

- 3- Habilite o sensor de luminosidade e seu Widget correspondente. E descreva o que aconteceu. Em seguida, tente alterar o valor da luminosidade para zero ☺.
- 4- O sensor de luminosidade possui um interpretador que define classes de interpretação em função do valor em lumens capturado pelo sensor. Habilite o interpretador e descreva o que acontece na aplicação móvel. Em seguida, modifique os valores das classes correspondentes de forma que o seu contexto atual seja “Luz forte”.
- 5- Vamos agora usar as informações geradas pelo interpretador para tomada de decisões. Na parte de tomada de decisão, habilite um trigger simples e defina => quando a classe de luminosidade for “Pouca Luz ou 20” a lanterna será ligada.
- 6- Modifique o contexto atual do smartphone para que a regra seja disparada. Em seguida, redefina as classes de interpretação de luminosidade para que a lanterna esteja sempre ligada.

- 7- Habilite o sensor do acelerômetro e descreva o que o seu widget e interpretador realizam como função. Para que serve o parâmetro de sensibilidade do interpretador?
- 8- Combine agora o acelerômetro e a luminosidade para a tomada de decisão. O que aconteceu no aplicativo móvel?
- 9- Habilite um agregador de contexto e modifique seus parâmetros para definir o contexto de alto nível "Perigoso". O que aconteceu na aplicação móvel? Modifique o seu contexto atual para que o contexto de alto nível Perigoso seja produzido.
- 10-Repita o processo para o contexto de alto nível "Seguro". O que aconteceu na aplicação móvel? Modifique o seu contexto atual para que o contexto de alto nível aconteça.
- 11-Utilize os contextos de alto nível para definir o trigger e modifique o seu contexto atual para que a lanterna seja acendida.
- 12-De acordo com as definições de Dey, após a última configuração do simulador Web, quais seriam os elementos que compõem o contexto atual do aplicativo móvel?
- 13-Quais seriam as zonas de observação e as zonas de interesse que o simulador oferece para a definição de contexto do aplicativo móvel?

6. **6. Eu acho que o ambiente apresenta muitas inconsistências. ***

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

7. **7. Eu imagino que as pessoas aprenderão como usar esse ambiente rapidamente. ***

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

8. **8. Eu achei o ambiente atrapalhado de usar. ***

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

9. **9. Eu me senti confiante ao usar o ambiente. ***

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

10. **10. Eu precisei aprender várias coisas novas antes de conseguir usar o ambiente. ***

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

Avaliação pedagógica

11. **11. Eu acho que o ambiente e sua proposta promovem debate sobre os tópicos trabalhados com alunos e/ou com o próprio professor. ***

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

12. **12. Eu acho que os recursos tecnológicos do Lucy servem como mediador no processo de aprendizagem dos conceitos trabalhados na disciplina. ***

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

13. **13. Eu acho que o nível de dificuldade apresentado nas atividades do ambiente é apropriado para mim (aluno). ***

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

Avaliação de interface

14. **14. O ambiente possui uma interface que torna seu uso mais agradável.**

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

15. **15. Os recursos interativos do ambiente me permitem alterar a sua configuração, de modo a obter respostas diferentes de acordo com minhas ações.**

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

16. **16. A todo momento fui capaz de identificar a seção do ambiente em que estava localizado.**

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

17. **17. Os ícones que dão acesso a outras seções do ambiente são facilmente reconhecíveis.**

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

18. **18. Há contraste suficiente entre fontes e fundo de tela, facilitando a leitura de textos.**

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

19. **19. As fontes utilizadas apresentam tamanho adequado.**

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

20. Há consistência visual na apresentação de informações e recursos gráficos.

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

Sugestões

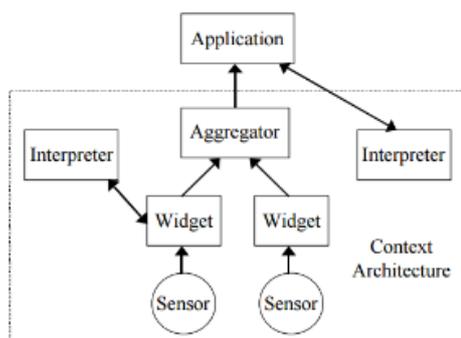
21. Caso possua alguma sugestão para o Ambiente Lucy, por favor, escreva no espaço abaixo.

APÊNDICE F – QUESTIONÁRIO GUIA DA PRÁTICA DE PROGRAMAÇÃO



Universidade Federal do Ceará
 Campus Fortaleza
 Centro de Ciências Exatas / UFC Virtual
 Departamento de Computação / Sistemas e Mídias Digitais

Prática sobre o Framework Context Toolkit



Context Toolkit Framework



Anind K. Dey and Gregory D. Abowd - 2000

Informações Iniciais

A prática de sala de aula, com o Ambiente Lucy, teve como objetivo mostrar/simular o fluxo de informações em aplicações sensíveis ao contexto conforme o framework proposto por Dey e Abowd. A simulação utilizada possuía camadas de sensores, widgets, interpretadores, agregadores e tomadas de decisão. Cada uma dessas camadas, com suas características específicas, segue princípios dos componentes do Context Toolkit Framework.

Essa proposta de atividade de codificação, possui um código inicial Android com estrutura semelhante à estrutura do framework presente no simulador trabalhado em sala de aula. Esse código é somente *mobile*, ou seja, não se conecta com nenhuma ferramenta Web, diferente da aplicação móvel usada em sala de aula.

O código está disponível em: <https://1drv.ms/f/s!AnfTpOzL2MeOhst8oLcCOQIS1RHLdA>

No anexo dessa prática, é possível visualizar um diagrama de classes que descreve a estrutura do código da prática

Etapa da prática- Observação

1- Compile o código Android do projeto da prática e o execute em seu *smartphone*. Observe as informações que aparecem. Adicione como resposta a essa questão *printscreen* das telas do seu celular que exibam os comportamentos listados a seguir.

Primeiro tem-se os elementos contextuais utilizados: acelerômetro, luminosidade, eixo do acelerômetro, nível de luminosidade e segurança do usuário.

Em seguida, tem-se o valor numérico do nível de luminosidade em lx e uma barra de progresso que mostra a intensidade da luz recebida pelo sensor do dispositivo e um texto de caracterização do nível corrente da luz: *none*, *low* e *high*.

Há ainda os valores para cada eixo do acelerômetro e uma imagem que indica em qual eixo está tendo maior aceleração.

Etapa da prática- Entendendo o código

2- Escolha três classes do projeto Android (dentre as mostradas no diagrama UML do anexo) e defina seu papel dentro da aquisição de contexto. Descreva em seguida, dois métodos de cada uma dessas classes explicando a funcionalidade por ele provida

3-No código inicial proposto pela prática, quais os elementos contextuais existentes? Quais atributos dessas classes poderiam ser definidos como propriedades contextuais de baixo nível (i.e., providas por sensores ou widgets) e quais seriam de alto nível (i.e., providas por agregadores e/ou interpretadores)? Explique.

Etapa da prática- Corrigindo o código

4- No código inicial do projeto Android, uma tomada de decisão foi codificada. Ela define que:
“Quando o nível de luz corrente for igual a 0 e a aceleração for maior em x, um popup é exibido mostrando que o usuário está em perigo”

No entanto, o indicador do nível de aceleração da aplicação está muito sensível e o popup, às vezes, aparece apenas com a aceleração no eixo x. Corrija esses dois problemas (indicador de aceleração sensível demais e popup aparecendo no momento errado) utilizando o código Android ContextPractice fornecido.

Etapa da prática- Estendendo o código

5. Finalizada a atividade anterior, acrescente um novo sensor à aplicação e crie seu correspondente *widget*, interpretador, agregador e uma tomada de decisão com um atuador.

Questionário de avaliação da prática: <https://goo.gl/forms/I7Os5iIKc7JNDAKs2>

APÊNDICE G – AVALIAÇÃO DA PRÁTICA DE PROGRAMAÇÃO

* Required

1. **1) Essa prática me ajudou a entender e conhecer melhor a arquitetura do framework Context Toolkit proposta por Dey e Abowd. ***

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

2. **2) Essa prática me ajudou a entender e conhecer melhor o fluxo de informações em aplicações sensíveis ao contexto baseadas na arquitetura Context Toolkit. ***

Mark only one oval.

	1	2	3	4	5	
Discordo Completamente	<input type="radio"/>	Concordo Completamente				

3. **3) Descreva o que mais você achou interessante na prática de programação. ***

4. **4) Descreva o que você mudaria ou acrescentaria na prática de programação. ***

APÊNDICE H – AVALIAÇÃO QUALITATIVA - LUCY

Detalhamento das experiências com UbiComp

* Required

1. Qual sua experiência com Computação Ubíqua? *

2. Qual sua experiência no Ensino de UbiComp? *

Satisfação com o Lucy

3. 1) Você considera aceitáveis os tipos de atividades propostas pelo Lucy? *

4. 2) Para qual público você considera ideal a utilização desse ambiente: graduandos e/ou pós-graduandos? *

5. 3) Você consegue visualizar a utilização do Lucy em suas disciplinas? *

6. 4) Que outros conceitos de UbiComp você usaria com uma abordagem semelhante a de Sensibilidade ao Contexto proposta por Lucy? *

7. 5) Que impressões você tem com relação à proposta do ambiente Lucy? *

8. 6) Você conhece ambientes/ferramentas com proposta similar a do Lucy? *

9. 7) Qual a forma mais adequada, na sua opinião, de avaliar a efetividade desse ambiente no ensino prático de conceitos de UbiComp? *

10. 8) O que você acrescentaria/melhoraria nesse ambiente? *

11. 9) Você concorda com a abordagem realizada nas aulas com o Lucy? *

12. 10) O que você acrescentaria/mudaria na abordagem das aulas que utilizam o Lucy? *
