



**UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE TELEINFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA**

PAULO DE TARSO CAVALCANTE PEQUENO FILHO

**MODELAGEM FORMAL, SIMULAÇÃO E ANÁLISE EM REDES DE PETRI
COLORIDAS DA REALIZAÇÃO DE ATIVIDADES LABORATORIAIS A
DISTÂNCIA PRESENTES NO *FRAMEWORK UBIQUITOUS LAB FOR PRACTICAL
ACTIVITIES (U-LABPA)***

FORTALEZA

2016

PAULO DE TARSO CAVALCANTE PEQUENO FILHO

MODELAGEM FORMAL, SIMULAÇÃO E ANÁLISE EM REDES DE PETRI
COLORIDAS DA REALIZAÇÃO DE ATIVIDADES LABORATORIAIS A DISTÂNCIA
PRESENTES NO *FRAMEWORK UBIQUITOUS LAB FOR PRACTICAL ACTIVITIES* (U-
LABPA)

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia de Teleinformática, (PPGETI) da Universidade Federal do Ceará, como requisito parcial à obtenção do título de Mestre em Engenharia de Teleinformática. Área de concentração: Sinais e Sistemas.

Orientador: Prof. Dr. Giovanni Cordeiro Barroso.

Coorientador: Prof. Dr. José Marques Soares.

FORTALEZA

2016

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)

P1m Pequeno Filho, Paulo de Tarso Cavalcante.
Modelagem formal, simulação e análise em Redes de Petri Coloridas da realização de atividades laboratoriais a distância presentes no framework ubiquitous lab for practical activities (u-LabPA) / Paulo de Tarso Cavalcante Pequeno Filho. – 2016.
78 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia de Teleinformática, Fortaleza, 2016.
Orientação: Prof. Dr. Giovanni Cordeiro Barroso.
Coorientação: Prof. Dr. José Marques Soares.

1. Computação Ubíqua. 2. Trabalhos Laboratoriais. 3. Redes de Petri Colorida. I. Título.

CDD 621.38

PAULO DE TARSO CAVALCANTE PEQUENO FILHO

MODELAGEM FORMAL, SIMULAÇÃO E ANÁLISE EM REDES DE PETRI
COLORIDA DA REALIZAÇÃO DE ATIVIDADES LABORATORIAIS A DISTÂNCIA
PRESENTES NO *FRAMEWORK UBIQUITOUS LAB FOR PRACTICAL ACTIVITIES* (U-
LABPA)

Dissertação apresentada ao Programa de Pós-
Graduação em Engenharia de Teleinformática
da Universidade Federal do Ceará, como
requisito parcial à obtenção do título de Mestre
em Engenharia de Teleinformática. Área de
concentração: Sinais e Sistemas.

Aprovada em: 16/12/2016.

BANCA EXAMINADORA

Prof. Dr. Giovanni Cordeiro Barroso (Orientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. José Marques Soares (Coorientador)
Universidade Federal do Ceará (UFC)

Prof. Dr. Gabriel Antoine Louis Paillard
Universidade Federal do Ceará (UFC)

Prof. Dr. Felipe Maia Galvão França
Universidade Federal do Rio de Janeiro (UFRJ)

Ao meu orientador, professor doutor Giovanni, pela paciência e dedicação que teve nesta minha jornada, ao meu amigo professor Wellington Wagner pelo incentivo e força que me deu para que eu fizesse este mestrado, a meu primo Paulo André por toda ajuda dedicada e, por último, e não menos importante, à minha esposa Lucíola Caminha pela paciência que teve comigo.

AGRADECIMENTOS

Ao Prof. Dr. Giovanni Cordeiro Barroso e Prof. Dr. José Marques Soares, pela excelente orientação.

Aos professores participantes da banca examinadora Prof. Dr. Gabriel Antoine Louis Paillard e Prof. Dr. Felipe Maia Galvão França pelo tempo, pelas valiosas colaborações e sugestões.

Aos colegas da turma de mestrado, pelas reflexões, críticas e sugestões recebidas.

RESUMO

A Computação Ubíqua é uma área que relaciona tecnologias e conceitos como Pervasividade, Mobilidade, Adaptabilidade e Aplicações Orientadas à Localização. Dentre as tecnologias que se destacam hoje nesta área, têm-se as Redes de Sensores Sem Fio e de Identificação por Radiofrequência, que estão sendo usadas em conjunto para monitorar e identificar objetos e pessoas. Dentro do escopo da Computação Ubíqua, existem os Laboratórios Ubíquos (u-Lab) que são laboratórios que possuem sensoriamento, provimento de serviços de *software* de apoio ao usuário, apoio a realização de atividades, dentre outras características. A fim de dar suporte às práticas realizadas em Laboratórios Ubíquos foi criado o *framework* chamado *Ubiquitous Laboratory for Practical Activities* (u-LabPA), que permite a alunos colherem dados de atividades laboratoriais e acessá-los a posteriori na forma de simulações ou ambientes virtuais. Neste trabalho é feita a modelagem formal e simulação em Redes de Petri Coloridas (RdPC), bem como a análise de resultados dos requisitos “Autenticação”, “Cadastra Atividade”, “Registra Atividade de Usuário”, “Disponibiliza Atividades de Usuário” e “Realiza Atividade”, presentes no *framework* u-LabPA. Os resultados desta análise mostram que o fluxo de realização de atividades do *framework* u-LabPA atendeu a seus requisitos básicos bem como aos testes de estresse para o qual foi submetido, mantendo sua coerência e o seu comportamento dentro do esperado.

Palavras-chave: Computação Ubíqua. Trabalhos Laboratoriais. *WSN*. *RFID*. RdPC.

ABSTRACT

Ubiquitous Computing is an area that relates technologies and concepts such as Pervasiveness, Mobility, Adaptability and Location-Oriented Applications. Among the technologies that stand out today in this area there are Wireless Sensor Networks and Radio Frequency Identification Networks, which are being used together to monitor and identify objects and people. Within the scope of Ubiquitous Computing, there are Ubiquitous Laboratories (u-Lab) which are laboratories that have sensing, provision of software services to support the user, support to the performance of activities, among other characteristics. In order to offer support to practices performed in Ubiquitous Laboratories a framework called Ubiquitous Laboratory for Practical Activities (u-LabPA) was created, which allows students to collect data from laboratory activities and to access them later in the form of simulations or virtual environments. In this work we perform the formal modeling and simulation using Colored Petri Nets (RdPC), as well as the analysis of the results of the requisits "Authentication", "Register Activity", "Register User Activity", "Provide User Activities", "Perform Activity", present in the u-LabPA framework. The results of this analysis reveal that the activity flow of the u-LabPA framework met its basic requirements as well as the stress tests for which it was submitted, maintaining its consistency and its behavior within the expected spectrum.

Keywords: Ubiquitous Computing. Laboratory Practices. WSN. RFID. RdPC.

LISTA DE FIGURAS

Figura 1-	Visão Geral u-LabPA.....	17
Figura 2 -	Diagrama de Atividades do Fluxo de Realização de Atividades por professor e por aluno	18
Figura 3 -	Evolução da Computação Ubíqua.....	21
Figura 4 -	Ilustração de uma RdP	25
Figura 5 -	Representação do disparo da transição t1.(a) estado inicial da RdP antes do disparo, (b) estado da RdP após o disparo.....	26
Figura 6 -	Representação de uma rede de Petri colorida	27
Figura 7 -	<i>Ubiquitous Classroom</i>	33
Figura 8 -	Visão geral do sistema u-LabPA.....	37
Figura 9 -	Casos de Uso de Autenticação e Realiza atividade.....	39
Figura 10 -	Laboratório Modelado.....	40
Figura 11 -	Modelo RdPC do u-LabPA	43
Figura 12 -	Requisitos: “Autenticação” e “Cadastra Atividade”	46
Figura 13 -	Modelo RdPC de Autenticação e Cadastra Atividade	47
Figura 14 -	Modelo RdPC de Alocação de Atividades.....	49
Figura 15 -	Modelo RdPC para Alocação de Instrumentos, Componentes e Bancada para usuário	51
Figura 16 -	Modelo RdPC do Laboratório Virtual para Experiências Simuladas	52
Figura 17 -	Requisito - Registra atividade de usuário.....	54

Figura 18 - Modelo RPC da Devolução de Instrumentos, Componentes e Bancada para	
Usuário.....	55

LISTA DE GRÁFICOS

Gráfico 1 - Estatísticas de entradas dos alunos no u-LabPA	58
Gráfico 2 - Estatísticas das saídas dos alunos logo após a entrada no u-LabPA	59
Gráfico 3 - Estatísticas das saídas dos alunos sem concluir a atividade	60
Gráfico 4 - Estatísticas das atividades retomadas pelos alunos	61
Gráfico 5 - Estatísticas das alocações	62
Gráfico 6 - Estatísticas das atividades finalizadas pelos alunos	62

LISTA DE ABREVIATURAS E SIGLAS

AVA	Ambiente Virtual de Aprendizagem
AVE	Ambiente Virtual de Exercícios
CGETI	Curso de Graduação em Engenharia de Teleinformática
DETI	Departamento de Engenharia de Teleinformática
EAD	Educação a Distância
MEC	Ministério da Educação
UFC	Universidade Federal do Ceará
U-LABPA	Ubiquitous Laboratory for Practical Activities
ALMA	Ambiente Laboratorial de Monitoramento de Atividades
SMIL	Sistema de Monitoramento e Identificação em Laboratório Real
SLOP	Sistema de Localização de Objetos e Pessoas
WSN	Wireless Sensor Networks
RFID	Radio-Frequency Identification
AAP	Ambiente de Atividades do Professor
AAA	Ambiente de Atividades do Aluno
LAVES	Laboratório Virtual para Experiências Simuladas
CPN TOOLS	Colored Petri Nets TOOLS
RDP	Redes de Petri
RDPC	Redes de Petri Coloridas

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Justificativa	18
1.2	Objetivos	19
1.3	Publicação	19
1.4	Organização do texto	19
2	REFERENCIAL TEÓRICO	21
2.1	Computação ubíqua	21
2.2	Computação ubíqua na educação	23
2.3	Especificação formal	24
2.4	Redes de Petri	24
2.5	Redes de Petri Coloridas	26
2.6	Monitores	28
2.7	Engenharia e definição de requisitos	28
2.8	Trabalhos relacionados	29
2.8.1	<i>Modelagem de interações utilizando modelos formais</i>	30
2.8.2	<i>Modelagem formal de requisitos de software utilizando Redes de Petri</i>	31
2.8.3	<i>Computação ubíqua</i>	32
2.8.4	<i>Análise comparativa dos trabalhos relacionados</i>	35
3	MODELAGEM FORMAL DO U-LABPA EM REDES DE PETRI COLORIDAS	36
3.1	Cenário proposto	40
3.2	u-LabPA – fluxo de atividades	41
3.3	Cenário geral para simulação com RdPC	42
3.3.1	<i>Seguem a descrição de cada um dos modelos criados para o fluxo “Realização de Atividades do u-LabPA”</i>	42
3.3.1.1	<i>Descrição do cenário</i>	42
3.3.1.2	<i>Descrição da RdPC</i>	43
3.3.2	<i>Modelo “Autenticação e Cadastra Atividade”</i>	45
3.3.2.1	<i>Descrição do cenário</i>	45
3.3.2.2	<i>Descrição da RdPC</i>	46
3.3.3	<i>Modelo “Visualiza e Abre Atividade”</i>	48
3.3.3.1	<i>Descrição do cenário</i>	48
3.3.3.2	<i>Descrição da RdPC</i>	49

3.3.4	<i>Modelo alocação de instrumentos, componentes e bancada para usuário/Realização e anotação da atividade</i>	50
3.3.4.1	<i>Descrição do cenário</i>	50
3.3.4.2	<i>Descrição da RdPC</i>	50
3.3.5	<i>Criação de uma instância virtual do laboratório</i>	51
3.3.5.1	<i>Descrição do cenário</i>	51
3.3.5.2	<i>Descrição da RdPC</i>	52
3.3.6	<i>Devolução de instrumentos, componentes e bancada para usuário</i>	53
3.3.6.1	<i>Descrição do cenário</i>	53
3.3.6.2	<i>Descrição da RdPC</i>	54
3.3.7	<i>Registra atividade de usuário</i>	54
4	ANÁLISE DE RESULTADOS	56
4.1	Estatísticas de entradas dos alunos no u-LabPA	58
4.2	Estatísticas das saídas dos alunos logo após a entrada no u-LabPA	58
4.3	Estatísticas das saídas dos alunos sem concluir a atividade	60
4.4	Estatísticas das atividades retomadas pelos alunos	61
4.5	Estatísticas das alocações e das atividades finalizadas pelos alunos	62
5	CONCLUSÕES	64
5.1	Trabalhos futuros	65
	REFERÊNCIAS	67
	APÊNDICE A – TABELA DE DADOS PROVENIENTES DOS MONITORES	72
	APÊNDICE B – DECLARAÇÕES REALIZADAS NO CPN TOOLS	73
	ANEXO A – FUNCIONALIDADES	75

1 INTRODUÇÃO

Cada vez mais dependentes de tecnologia, pertencemos a uma sociedade formada por uma grande rede de usuários interconectados. As Tecnologias Digitais abriram um leque de possibilidades cada vez maior nesses últimos anos e uma das grandes beneficiadas é a educação. É cada vez mais comum o uso de dispositivos móveis na educação, como *tablets*, *smartphones*, *notebooks*, dentre outros, e de acordo com Moran (2013, p.30).

As tecnologias móveis desafiam as instituições a sair do ensino tradicional em que o professor é o centro, para uma aprendizagem mais participativa e integrada, com momentos presenciais e outros com atividades a distância, mantendo vínculos pessoais e afetivos, estando juntos virtualmente.

O uso de tecnologia está cada vez mais presente nas Instituições de Ensino, seja nas atividades de gestão escolar ou em laboratórios. Apesar das facilidades no acesso às tecnologias, muitos professores ainda não se sentem à vontade no seu manuseio, necessitando de capacitação. O uso das novas tecnologias permite mudanças que viabilizam ações até pouco tempo inviáveis, quebrando barreiras associadas a espaço e tempo. Atualmente escolas e universidades têm recebido com frequência recursos tecnológicos, tornando o espaço educacional mais atrativo, interativo e dinâmico, requerendo a devida qualificação dos professores.

As novas Tecnologias Digitais de Comunicação são desafios presentes na rotina do professor, pois antes esse sujeito assumia seu papel de forma presencial, dentro de um ambiente confinado, mas isso mudou. Nesse contexto, vale destacar dois conceitos importantes: ubiquidade e mobilidade. Mobilidade é o termo que identifica dispositivos que podem ser operados sem fio ou a distância, tendo como vantagem permitir o acesso a informações de qualquer lugar, em qualquer tempo. A ubiquidade é responsável por tornar a interação homem-computador invisível, prover de forma natural essa comunicação. Internet das coisas e dispositivos vestíveis podem ser utilizados como exemplos de uso de tecnologia ubíqua sem percepção pelo lado do usuário. Esses conceitos estão intimamente ligados quando se fala em tecnologias móveis, operadas à distância de modo contínuo e livre através da internet.

A Computação Ubíqua (WEISER,1993) desenvolve novas formas de pensar os sistemas computacionais, unindo-os aos objetos e permitindo que estes objetos interajam com os usuários para fornecer serviços desejados.

Peixoto, Carvalho e Sarmiento (2016), cita características fundamentais que descrevem o uso de Computação Ubíqua, são elas:

Adaptabilidade: as aplicações podem se reconfigurar dinamicamente para atender a variações do ambiente ou redefinição de requisitos;

Descentralização: capacidade de delegar entre os dispositivos funções que irão desempenhar em determinado ambiente;

Descoberta automática de serviços: serviços existentes no ambiente são detectados automaticamente pelos dispositivos dos usuários permitindo inclusive configurações personalizadas;

Escalabilidade: capacidade de o sistema manter a plenitude dos seus serviços mesmo com um grande número de dispositivos de usuários utilizando suas funções;

Heterogeneidade: capacidade de o sistema suportar uma grande variedade de dispositivos e serviços sem comprometer seu funcionamento;

Interoperabilidade: comunicação entre os dispositivos sem intervenção humana;

Invisibilidade: capacidade dos sistemas estarem ativos e desempenharem funções sem que os usuários percebam;

Onipresença: capacidade dos serviços/dispositivos migrarem de tecnologia sem que para isso o usuário tenha que intervir ou ter conhecimento dessa mudança;

Sensibilidade ao contexto: capacidade da aplicação identificar situações do ambiente que podem influenciar seu comportamento.

Tolerância a falhas: característica de identificar possíveis falhas e manter serviço ativo sem que o usuário perceba.

Através da difusão dos novos dispositivos de informação e comunicação, como *smartphones*, *tablets*, sensores e outros equipamentos com alto poder de comunicação e processamento disponíveis, tornou-se possível a Computação Ubíqua (PEIXOTO; CARVALHO; SARMENTO, 2016).

Duas das tecnologias utilizadas no contexto de Computação Ubíqua se destacam perante a comunidade devido ao grande número de estudos e aplicações desenvolvidas, são

elas: Redes de Sensores Sem Fio (RSSF ou WSN, do inglês, *Wireless Sensor Networks*) e Identificação por Rádio Frequência (RFID, do inglês *Radio-Frequency Identification*). Elas permitem tanto a localização de objetos quanto seu monitoramento por parte de sistemas computacionais.

Neste contexto, Sarmiento (2016), propõe o *framework Ubiquitous Laboratory for Practical Activities* (u-LabPA). O u-LabPA permite a alunos colherem dados de seus experimentos feitos em laboratórios e acessá-los posteriormente na forma de simulações em ambientes 2D ou de Realidade Virtual. A proposta é que professores de cursos na modalidade de Educação a Distância se beneficiem desta tecnologia, pois podem monitorar as atividades de seus alunos e acompanhá-las mesmo estando geograficamente afastados. Este *framework* também suporta acesso por dispositivos móveis, permitindo maior facilidade para seu uso em ambiente laboratorial. Note que este trabalho de modelagem é feito sobre o *framework* u-LabPA e aplicações construídas com ele, sendo, portanto, restrito ao *software*. O escopo da modelagem não abrange fatores do laboratório real e sim como o *software* percebe este laboratório para as atividades. Assim, o formalismo pode identificar realmente problemas dentro dos requisitos e do fluxo de atividades do *framework*.

O u-LabPA permite também o uso de uma rede mista de sensores sem fio e identificação por radiofrequência para mapeamento de equipamentos utilizados em laboratórios (e.g. Física, Química etc.) a fim de permitir que alunos portando dispositivos móveis como *smartphones* e *tablets* possam identificar tais equipamentos e monitorar seus experimentos. Assim, um aluno que esteja estudando e fazendo seus primeiros experimentos neste laboratório, pode obter informações específicas de um dado equipamento usando um leitor de RFID e monitorar algumas variáveis físicas (e.g.: tensão, resistência elétrica) de seus experimentos, usando a rede de sensores presente no laboratório. Embora dê suporte a WSN e RFID, o *framework* pode funcionar sem estas tecnologias ou mesmo suportar outras formas de identificação de objetos, como é o caso de códigos de barra bidimensionais. No entanto, seu maior potencial, que o faz estar inserido fortemente dentro da área de Computação Ubíqua, está no uso de WSN e RFID. Desta forma, o aluno pode realizar sua prática em um laboratório real, obter os dados para posterior reflexão sobre os resultados, além de discutir com outros alunos e professores o processo da prática e o experimento efetuado. É possível, ainda, utilizar os dados obtidos para alimentar simulações.

O escopo geral do *framework* abrange um conjunto de fluxos de atividades, dependendo do perfil escolhido conforme pode ser visto na Figura 1. O *framework* dá suporte

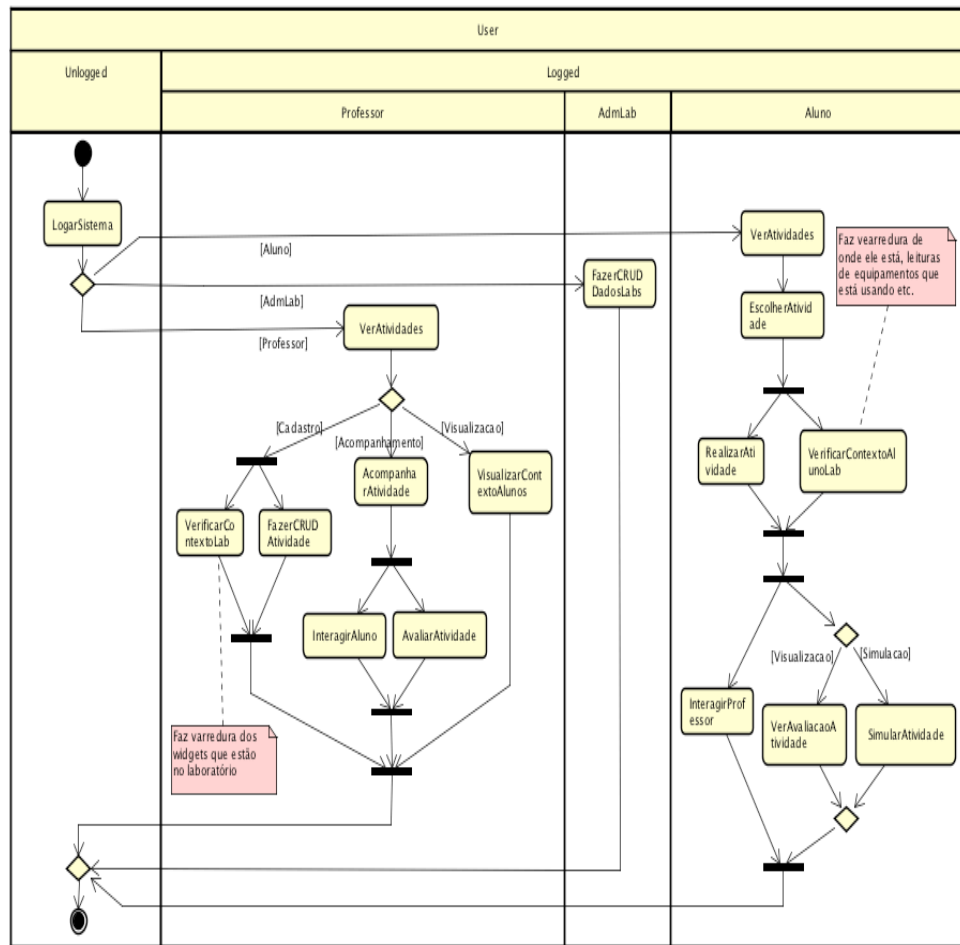
a perfis para que se possam agrupar as funcionalidades do sistema conforme o tipo de usuário. Assim, papéis dentro de uma aplicação feita com o uLabPA ficariam mais claros e organizados. Destes fluxos, destacam-se: Criação e Acompanhamento de Atividades; Realização de Atividades; Levantamento dos Dados de Laboratório. Em termos de escopo de requisitos¹, o *framework* está especificado nas seguintes funcionalidades:

- a. Autenticação;
- b. Cadastra usuário;
- c. Atribui perfil a usuário;
- d. Cadastra laboratório;
- e. Cadastra Atividade;
- f. Verifica Situação de Laboratório;
- g. Identifica Contexto de Aluno;
- h. Registra Atividade de Usuário
- i. Disponibiliza Atividade de Usuário;
- j. Realiza Atividade
- k. Lê *widget*;
- l. Atua sobre *widget*;
- m. Disponibiliza Dados de Atividade de Usuário.

Neste trabalho é feita a modelagem formal e simulação em Redes de Petri Coloridas (RdPC), bem como a análise de resultados dos requisitos “Autenticação”, “Cadastra Atividade”, “Registra Atividade de Usuário”, “Disponibiliza Atividades de Usuário” e “Realiza Atividade”, presentes no *framework* u-LabPA. Na Figura 2 é mostrado o fluxo de atividades que trata da “Realização de Atividades”, que faz parte do *framework*.

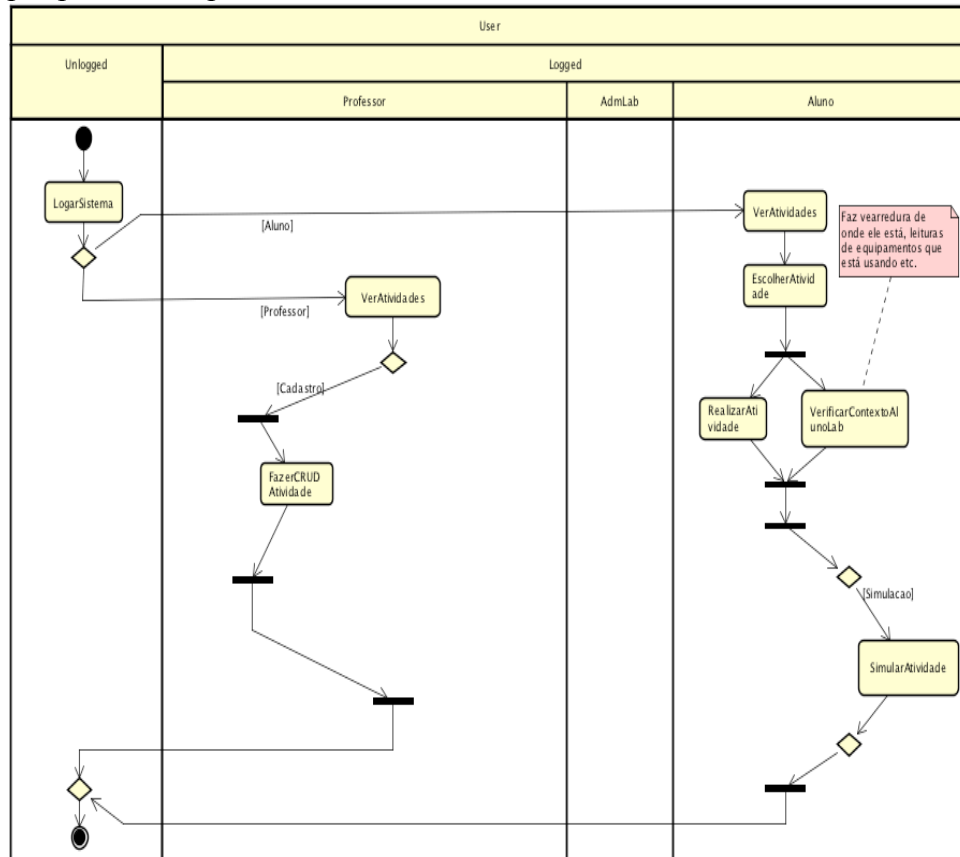
¹ Requisitos são as funções e restrições de um Sistema (SOMERVILLE, 2011)

Figura 1- Visão Geral u-LabPA



Fonte: Sarmento (2016)

Figura 2 - Diagrama de Atividades do Fluxo de Realização de Atividades por professor e por aluno



Fonte: Sarmento (2016)

1.1 Justificativa

Com a popularização da Educação a Distância no Brasil e seu uso crescente no Ensino Superior (ARANTES, 2011), fica premente a necessidade de soluções que possam abranger atividades práticas laboratoriais para cursos como Física, Química, Engenharias, etc. As soluções de Educação a Distância (EaD) dão resposta para os conteúdos trabalhados, as interações, diálogos, atividades que seriam trabalhadas em sala de aula e que podem ser vistas através de Ambientes Virtuais de Aprendizagem (AVA) (BARBOSA, 2005 *apud* SARMENTO, 2007). No entanto, as práticas laboratoriais não têm uma solução adequada ou próxima desta para permitir uma interação aluno-professor-prática. E nem tampouco se vê o uso das tecnologias para extrapolar as práticas tradicionais, permitindo uma revisita dos alunos a seus experimentos através de ambientes virtuais. Entretanto, encontram-se propostas que permitem a execução das práticas a distância ou seu monitoramento a distância (SILVA et al., 2016). Uma integração do controle, monitoramento e revisita das práticas em um AVA

permite uma consolidação do processo de aprendizagem teórico com o prático, sendo, portanto, importante para o aluno e essencial para a prática docente.

1.2 Objetivos

O objetivo principal deste trabalho é a modelagem formal em Redes de Petri Coloridas dos requisitos de “Autenticação”, “Cadastra Atividade”, “Registra Atividade de Usuário”, “Disponibiliza Atividades de Usuário” e “Realiza Atividade”, presentes no *framework* u-LabPA.

Os objetivos específicos desse trabalho são:

- Pesquisar soluções disponíveis atualmente, analisando suas características, visando aprimorar o apoio a atividades educacionais;
- Validar e verificar os requisitos de “Autenticação”, “Cadastra Atividade”, “Registra Atividade de Usuário”, “Disponibiliza Atividades de Usuário” e “Realiza Atividade” do *framework* u-LabPA utilizando em Redes de Petri Coloridas;
- Executar a simulação, o monitoramento e a análise dos dados provenientes da modelagem formal para validar os requisitos do sistema.

1.3 Publicação

O trabalho descrito nesta dissertação foi apresentado, em abril de 2016, no 8th *Euro American Conference on Telematics and Information Systems* (EATIS), intitulado **u-LabPA: A framework based on ubiquitous computing for laboratory activities** e está indexado através do IEEExplorer e ACM Library por meio do DOI 10.1109/EATIS.2016.7520146. (PEQUENO FILHO et al. 2016).

1.4 Organização do texto

A dissertação está organizada em cinco capítulos da forma como se segue. No Capítulo 2 são discutidas as bases teóricas associadas ao u-LabPA em um contexto

tecnológico e educacional, assim como são apresentadas técnicas e tipos de filtragem incluindo a demonstração de sistemas já existentes no contexto educacional; também é discutida neste capítulo a técnica de modelagem formal utilizada no presente trabalho; por fim, são mostrados trabalhos relacionados. No Capítulo 3 são apresentadas as modelagens formais em Redes de Petri Coloridas referentes à “Autenticação”, “Cadastra Atividade”, “Registra Atividade de Usuário”, “Disponibiliza Atividades de Usuário” e “Realiza Atividade”. No Capítulo 4 é feita a análise dos dados provenientes da modelagem, e simulação. Finalmente, no Capítulo 5 são apresentadas as conclusões deste trabalho, bem como as perspectivas e sugestões para trabalhos futuros.

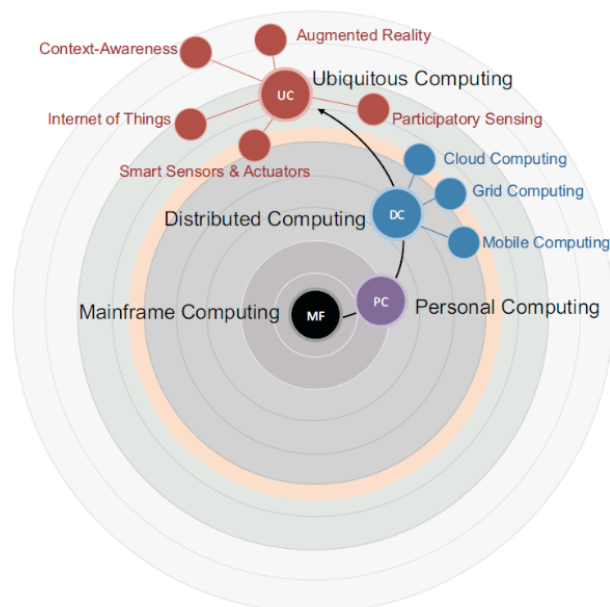
2 REFERENCIAL TEÓRICO

A seguir são abordados conceitos de computação ubíqua, sua aplicação na educação e Redes de Petri Coloridas, bem como requisitos de *software* e trabalhos relacionados à presente pesquisa.

2.1 Computação ubíqua

Como ilustrado na Figura 3, a Computação Ubíqua é fruto da evolução de alguns outros paradigmas que tiveram início com a computação de mainframe (PEIXOTO; CARVALHO; SARMENTO, 2016). Numa primeira fase, os mainframes ocupavam o cenário da época com grande processamento de informação centralizado. Posteriormente, veio a segunda fase, representada pela computação pessoal, conhecida pelos computadores de pequeno porte e de baixo custo, destinados exclusivamente ao uso pessoal. Em seguida, temos a terceira fase ou a era da computação distribuída. Esta se caracterizou pela dispersão de componentes de hardware e de software que se comunicam e executam como um único sistema. Então, na quarta fase ou fase atual, temos a concretização da Computação Ubíqua.

Figura 3 - Evolução da Computação Ubíqua



Fonte: Knappmeyer (2013)

A Computação Ubíqua (no inglês, *Ubiquitous Computing – UbiComp*) é uma intercessão entre a Computação Pervasiva (no inglês, *Pervasive Computing - PerComp*) e a Computação Móvel (no inglês, *Mobile Computing - MobComp*) (ARAÚJO, 2003). Outros aspectos pertinentes à computação ubíqua são o grau de embarcamento de sistemas, a adaptabilidade e a localização.

O conceito de computação pervasiva implica que o computador está embarcado ao ambiente de forma invisível para o usuário, tendo a capacidade de obter informações acerca do ambiente circundante e utilizá-las para controlar, configurar e ajustar a aplicação para melhor se adequar às características do ambiente. O ambiente também pode e deve ser capaz de detectar outros dispositivos que adentrem a ele (CIRILO, 2007).

A mobilidade diz respeito à capacidade de um sistema oferecer serviços mesmo que um usuário esteja em movimento - a Computação Móvel é a área da computação que trata destes sistemas. Sua aplicação na Educação pode ser vista em Paillard et al. (2012). A adaptabilidade é a capacidade de um sistema reagir ao seu contexto. Um contexto é qualquer informação que pode ser usada para caracterizar a situação de uma pessoa, objeto ou lugar (BARBOSA et al. 2008). Assim, uma aplicação adaptável responderia ao seu contexto e teria suas características modificadas para a melhor adequação ao mesmo ou forneceria as informações solicitadas conforme o contexto em que estivesse. Por fim, como característica da Computação Ubíqua, tem-se a localização. Este conceito diz respeito à construção de sistemas que levem em consideração a localização do equipamento utilizado, bem como o usuário que o está portando. Um exemplo do uso de Localização é permitir saber o conjunto de restaurantes próximos ao local onde uma pessoa se encontra.

A popularização dos dispositivos móveis permite que se possa ter um vislumbre dos cenários ubíquos, como os vistos em Araújo (2003). Esses dispositivos também possibilitam a criação de uma rede, onde tarefas, atividades e problemas computacionais podem ser repartidos, a fim de facilitar a resolução de problemas. No entanto, não somente os dispositivos móveis são importantes, Redes de Sensores e Identificadores de Rádio Frequência são tecnologias que podem ser usadas para identificar usuários e localizá-los em um dado ambiente, bem como para colher informações deste ambiente. Isso produz contextos que permitem criar regras para a ação dos sistemas ubíquos (SARMENTO et al., 2012).

2.2 Computação ubíqua na educação

O processo de ensino-aprendizagem requer esforços didáticos e metodológicos sempre novos e instigantes. Para Vygotsky o processo de aprendizagem é uma atividade social, portanto, deve ser feita em um âmbito cooperativo e interativo (OLIVEIRA, 2010). Este se relaciona com a visão de Piaget no referente ao caráter interacionista do processo de aprendizagem, onde o objeto e o sujeito devem participar de um processo de interação de mútua interferência (OLIVEIRA, 2010). Em ambientes computacionais ubíquos a interação é uma das peças chave, o que vem ao encontro da interatividade proposta por Piaget e Vygotsky.

Conceitos de Computação Ubíqua são vistos em diferentes disciplinas nas universidades, tais como Redes Móveis Sem Fio, Sistemas Distribuídos, Interação Humano Computador, Internet das Coisas ou mesmo naquelas dedicadas exclusivamente ao ensino de Computação Ubíqua.

Chalmers (2015), descrevem um cenário de organização de disciplinas que possuem conceitos e práticas de Computação Ubíqua na universidade de Sussex, no Reino Unido. Essas aulas estão organizadas em três partes: exposição teórica, seminários e práticas em laboratório.

Na exposição teórica, são apresentados os conceitos fundamentais de Computação Ubíqua e trabalhos que abrangem, de forma mais generalizada, o estado da arte relacionado ao tema. No momento da apresentação de seminários, o foco é analisar criticamente as teorias e as soluções existentes para resolução de problemas relacionados aos conceitos ubíquos. Nas práticas em laboratório, o objetivo é programar variados dispositivos, para justificar o entendimento do que foi visto na teoria. O foco é a concepção, a execução e a análise de experimentos, apresentando, em seguida, os resultados obtidos.

Oliveira e Costa (2014), apresentam uma experiência prática do uso de computação pervasiva em sala de aula, em que dispositivos computacionais foram conectados por rede sem fio para proporcionar a interação entre os alunos, proporcionando uma maior dinamicidade na turma.

Nesta dissertação desenvolveu-se um conjunto de modelos em Redes de Petri Coloridas, representando o fluxo de realização de atividades em um laboratório genérico, tendo como base o *framework* u-LabPA. A partir desses modelos, foi possível a simulação e

análise dos dados a fim de validar o fluxo e os requisitos básicos do *framework*. Como instrumentos para coleta de dados, foram utilizados monitores. Maior detalhamento será apresentado no decorrer do trabalho.

2.3 Especificação formal

Sommerville (2011), afirma que especificações formais se fazem necessárias para analisar, descrever e verificar ações e comportamentos de sistemas. Após o processo de levantamento de requisitos, a especificação formal pode ser utilizada na etapa de modelagem, de forma a retratar os fluxos de informação.

Segundo Woodcock et al. (2009), os métodos de modelagem formais permitem a análise e a verificação de modelos durante todo o ciclo de vida do projeto. Deste modo, tais métodos asseguram que o tempo gasto durante o processo de especificação formal seja compensado por um produto de maior qualidade, o que pode reduzir a quantidade de erros e a possibilidade de retrabalho posterior.

O uso de métodos formais permite incorporar maior confiabilidade aos processos que antecedem a etapa de codificação de uma aplicação (WOODCOCK et al., 2009). Dentre os métodos formais existentes e aplicáveis à modelagem da interatividade, as Redes de Petri propõem um formalismo de alto nível de abstração, capaz de representar os estados alcançáveis de um sistema a partir de uma ou mais condições, permitindo, também, a realização de tratamentos temporais e a descrição de mudanças que as transições causam aos estados do sistema (GEHLOT; NIGRO, 2010).

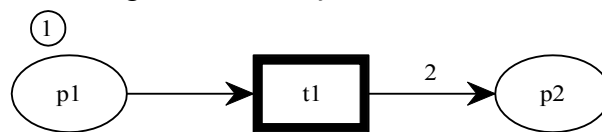
2.4 Redes de Petri

As Redes de Petri (RdP) são compreendidas como grafos bipartidos, ou seja, diagramas cujos elementos formam dois conjuntos disjuntos que não possuem unidades em comum (CARDOSO e VALETTE, 1997). Por se tratarem de um método formal baseado em modelos, as RdP são construídas utilizando-se abstrações com fundamentos matemáticos – tais como elementos e conjuntos –, além de operações capazes de definir a maneira e a sequência que as atividades são realizadas no sistema modelado.

Os elementos que compõem uma RdP são denominados lugares, transições, arcos e fichas (PETERSON, 1981). Os lugares – representados por círculos – simbolizam os estados de um sistema. As transições – que podem ser retângulos ou barras verticais – ilustram os eventos que podem ocorrer. Os arcos – retratados por linhas que ligam os lugares às transições e vice-versa – exemplificam os fluxos que acontecem no sistema e as fichas – expressas por pontos localizados dentro dos lugares – são indicadores que verificam se as condições associadas aos lugares são cumpridas.

Fernandes (2012), ressalta que a possibilidade de representar processos concorrentes é uma característica importante das RdP. Outra questão importante é a representação do comportamento dinâmico dos sistemas modelados (CARDOSO e VALETTE, 1997) que não são observados em outras linguagens, como, por exemplo, *Unified Modeling Language* (UML). Além disso, por ser uma linguagem formal, proporciona recursos para a verificação da correteude do sistema modelado.

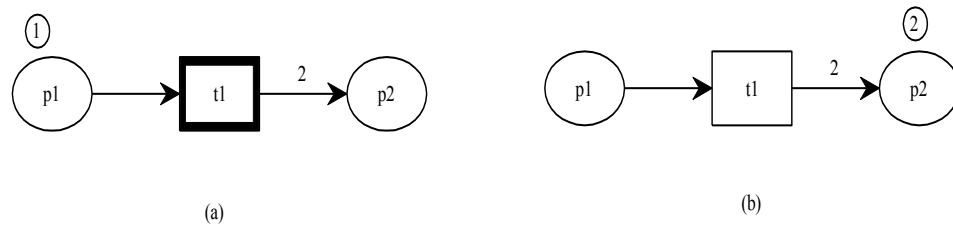
Figura 4 - Ilustração de uma RdP



Fonte: elaborado pelo autor.

A RP apresentada na Figura 05 ilustra a regra de disparo através da modelagem do comportamento dinâmico de um sistema e sua evolução. A Figura 05(a) apresenta o estado inicial do sistema. A mudança de estado, que acontece através do disparo da transição t1, é apresentada na Figura 05(b). Na Figura 05(a), existe uma ficha no lugar de entrada p1 e nenhuma ficha no lugar de saída p2. A marcação da rede é $M_0 = (1, 0)$. Nesta marcação a transição t1 está habilitada e pode disparar. Conforme apresentado na Figura 05(b), no disparo da transição t1, uma ficha é removida do lugar de entrada p1 e duas fichas são adicionadas ao lugar de saída p2, originando uma nova marcação ou estado do sistema $M_1 = (0, 2)$. Como pode ser observado neste exemplo, a quantidade de fichas removidas do lugar de entrada e adicionada ao lugar de saída depende do peso dos arcos.

Figura 5 - Representação do disparo da transição t1.(a) estado inicial da RdP antes do disparo, (b) estado da RdP após o disparo.



Fonte: elaborado pelo autor.

2.5 Redes de Petri Coloridas

Definidas como uma extensão das RdP clássicas, as Redes de Petri Coloridas (RdPC) são um formalismo de alto nível de abstração capaz de representar os estados que um sistema pode alcançar a partir de uma ou mais condições simultâneas (JENSEN, 2010). Permitem, também, a realização de tratamentos de caráter temporal e a descrição das mudanças que as transições podem causar aos estados de sistemas (GEHLOT; NIGRO, 2010). As RdPC têm sido aplicadas a uma ampla gama de aplicações, tais como em projetos de *hardware* (KIM et al., 2012), *software* (MARTINIE; NAVARRE; PALANQUE, 2014), análise de protocolos de rede (CHOOSANG; GORDON, 2014), sistema de arquivos paralelos (NGUYEN; APON, 2012) e sistemas distribuídos (WEIDLICH; MENDLING; GAL, 2013).

O uso das RdPC permite criar uma abstração capaz de compreender o projeto, a especificação e a validação de *hardware* e *software*. Assim como as RdP, elas também são representadas por grafos que possuem lugares, transições, arcos e fichas; todavia, nas RdPC é possível atribuir cores (ou tipos) a cada uma das fichas o que permite uma distinção das mesmas – e suas transições podem representar hierarquias, concedendo uma melhor organização ao sistema modelado (JENSEN, 2010).

Em RdPC as fichas são associadas a cores que permitem ser distinguidas e como consequência, a cada lugar se associa o conjunto de cores das fichas que podem pertencer a este lugar (CARDOSO e VALETTE, 1997). Além disso, é necessário associar às transições, determinadas operações denominadas funções de guarda, capazes de habilitar ou não a transição para a qual foram declaradas, e atribuir aos arcos expressões que ilustram as fichas que podem ser carregadas através dos modelos gerados.

Uma vantagem de se utilizar as RdPC para a modelagem de sistemas refere-se à compactação obtida nos modelos que são criados por meio dessa linguagem. Ao se empregar outras abordagens como, por exemplo, as RdP clássicas para realizar especificações formais, mostra-se necessário utilizar diversos lugares diferentes para representar fichas de origens diversificadas. Com as RdPC, torna-se possível usar apenas um lugar e especificá-lo com um conjunto de tipos capaz de comportar várias fichas ao mesmo tempo.

Conforme Jensen (2010), a definição formal de uma Rede de Petri Colorida é representada por uma eneátupla (9-tupla) $RdPC = (\Sigma, P, T, A, N, C, G, E, I)$, na qual:

Σ Denota um conjunto finito e não nulo de conjuntos de cores ou tipos;

P representa um conjunto finito de lugares;

T representa um conjunto finito de transições;

A representa um conjunto finito e disjunto de arcos;

N denota as funções de nó;

C representa os conjuntos e multiconjuntos de cores;

G representa as funções de guarda;

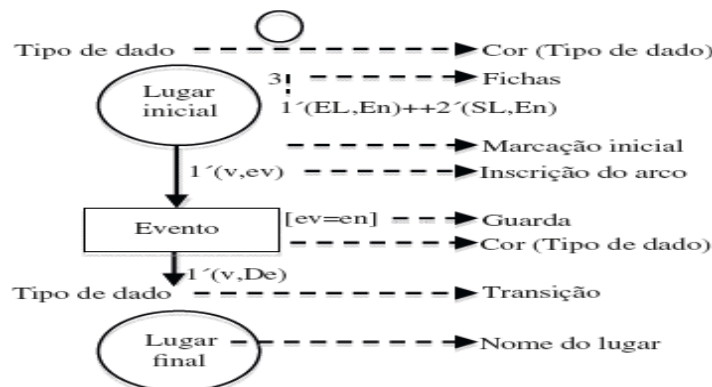
E representa as expressões de arcos (ou funções de expressão);

I é uma função de inicialização (marcação inicial).

O formato de apresentação dos modelos em RdPC facilita a compreensão de sistemas – já que eles se assemelham a diagramas de fluxos de dados – e isso permite uma melhor aceitação de equipes de projeto e desenvolvimento.

A Figura 06 mostra a estrutura principal de um modelo RdPC, constituído por lugares, transições e arcos. Com RdPC, é possível usar os tipos de dados e manipulação de dados complexos.

Figura 6 - Representação de uma rede de Petri colorida



Fonte: Araújo (2014)

2.6 Monitores

A utilização de Monitores neste trabalho foi importante para coleta de informações dos dados de saída da simulação, análise e exibição em forma gráfica. Na documentação eletrônica da ferramenta CPNTools², define-se monitor como: mecanismo que é usado para observar, fiscalizar, controlar ou modificar uma simulação de uma RdPC. Diferentes monitores podem ser definidos para uma dada rede. Os monitores podem inspecionar tanto as marcações de lugares como os elementos de ligação que ocorrem durante uma simulação. Eles também podem tomar as medidas apropriadas baseados nas observações feitas.

2.7 Engenharia e definição de requisitos

Em um sistema, as funções e restrições são seus Requisitos (SOMMERVILLE, 2011). Existem requisitos funcionais, não funcionais e de domínio. Os funcionais são declarações de funções de que um sistema deve fornecer, bem como suas reações a entradas específicas e como se comporta em determinadas situações. Os não funcionais são restrições sobre serviços ou funções oferecidas pelo sistema. Por fim, os de domínio são aqueles que se originam do “domínio de aplicação do sistema e que refletem características desse domínio” (SOMMERVILLE, 2011). O Levantamento de Requisitos é uma das fases fundamentais do processo de criação de um sistema de *software*, e envolve diversos atores, como analistas, clientes, usuários, entre outros. Podem ocorrer erros na interpretação dos requisitos, no número de requisitos ou na prioridade dos requisitos (FERNANDES, 2012). Para Sommerville (2011), a Engenharia de Requisitos é o processo de descobrir, analisar, documentar e verificar serviços e restrições. Ela estaria, portanto, relacionada diretamente com os processos de levantamento de requisitos. Para que os engenheiros de requisitos possam trabalhar é preciso que se possam extrair de forma eficaz e eficiente os requisitos dos atores diretamente envolvidos com o uso do sistema. Para Fischer e Silva (2001) (2001), há quatro etapas dentro da Engenharia de Requisitos para que se possa gerar ao final o Documento de Requisito de *Software*:

² (<http://cpntools.org/documentation/tasks/performance/monitors/start>)

- Extração de requisitos: refere-se a agrupar as informações a fim de verificar o que o usuário está solicitando;
- Especificação: é um documento formal que registra todas as informações obtidas na elicitação de requisitos.
- Verificação: é onde se procura assegurar a consistência dos dados, evitando conflitos entre requisitos, e;
- Validação: assegura que o documento descreve com precisão o sistema que o cliente deseja.

As Redes de Petri Coloridas podem ser usadas nas etapas de verificação e validação dos requisitos. No primeiro caso, elas podem verificar se o fluxo de dados e os requisitos apresentados seguem o caminho esperado, podendo ser vistos através de simulação, comportamentos que sejam anômalos a estes. O trabalho de Fernandes (2012), por exemplo, usa uma variação da RdPC para verificação dos requisitos. No segundo caso, dado que se tenham os requisitos bem documentados, e sua especificação esteja clara, pode ser feita a validação usando Redes de Petri Coloridas. Como exemplo, pode ser citado o trabalho de Fernandes (2012), novamente.

No processo de Elicitação de Requisitos, ou seja, de tornar explícitos os requisitos, algumas técnicas são propostas e utilizadas hoje. Segundo Fernandes (2012), são elas: Casos de uso, Diagrama de Fluxo de Dados, Processador de Requisitos de Linguagem, Metodologia de Engenharia de Requisitos de Software.

O presente trabalho usa como base Casos de Uso para a definição do modelo funcional, por estes já terem sido criados para o u-LabPA quando no seu processo de concepção inicial (SARMENTO, 2016).

2.8 Trabalhos relacionados

Nesta seção, são descritos trabalhos que envolvem modelagem de interações com o uso de Redes de Petri e trabalhos que utilizam Computação Ubíqua e RFID no contexto educacional, enfatizando suas características.

2.8.1 Modelagem de interações utilizando modelos formais

O *LOCPN* (*Learning Objects Production with Colored Petri Nets*) é um modelo também baseado em RdPC desenvolvido para guiar o processo de desenvolvimento de Objetos de Aprendizagem (OA) e auxiliar no desempenho de práticas pedagógicas (SOUZA et al., 2007). Um OA é um recurso digital desenvolvido para ser usado em ambientes educacionais, como um utensílio para facilitar o processo de aprendizagem dos usuários. Desse modo, os autores afirmam que incorporando o *LOCPN* aos passos de construção de OA alcança-se uma série de benefícios, como especificações mais intuitivas e facilidade de se encontrar erros. No trabalho, a aplicabilidade do *LOCPN* é validada por meio de seu uso durante a formalização de um OA. Os autores afirmam que a clareza da especificação traz uma redução de tempo considerável ao final do processo de criação do artefato. Entretanto, tal método mostra-se aplicável exclusivamente ao desenvolvimento de OA.

A ICO (*Interactive Cooperative Objects*) é uma linguagem de descrição de interfaces baseada em Redes de Petri Objeto³, utilizada para a engenharia e desenvolvimento de aplicações interativas (MARTINIE; NAVARRE; PALANQUE, 2014). A linguagem possui uma ferramenta CASE exclusiva intitulada PetShop (*Petri Net Workshop* ou, em tradução livre, Oficina de Redes de Petri), que é capaz de fornecer suporte para os processos de especificação, prototipação e validação de softwares com fluxos interativos. Além de compartilhar dos benefícios de uma base concreta numa notação formal de descrição, a ICO também faz uso de alguns conceitos emprestados do paradigma de orientação a objetos, como instanciação dinâmica, encapsulamento, classificação e herança. Para uma correta utilização desse paradigma, um dialeto híbrido das RdP precisou ser empregado a fim de compreender todo o escopo da orientação a objetos. Por causa disso, uma série de requisitos – não apenas em projeto de interfaces, como também em programação orientada a objetos, Java, implementação de sistemas e modelagem baseada em RdP – são necessários para representar corretamente as aplicações interativas ao se fazer uso da ICO, o que inevitavelmente induz uma série de curvas de aprendizagem nas diversas áreas em que os profissionais precisam obter domínio para tornarem-se aptos a trabalhar em projetos utilizando a linguagem.

³ Uma Rede de Petri que pode conter outra, na forma de Ficha, introduzindo o conceito de Redes de Petri aninhadas que se comunicam ao sincronizar transições em diferentes níveis (WIKIPEDIA, 2016).

Ciardelli, Bixio e Regazzoni (2011), propõem um modelo cognitivo para representar interações entre motoristas e veículos, com o intuito de considerar o comportamento dos usuários para detectar situações potenciais de perigo. O trabalho apresenta uma abordagem para analisar e modelar aplicações automotivas e explora uma plataforma de simulação junto a uma rede de sensores instalada num veículo. Por fim, resultados preliminares são apresentados para demonstrar a viabilidade do método e discussões apresentam que pesquisas continuadas nessa área são capazes de trazer melhorias para aplicações com enfoque na segurança.

Ma et al. (2010), apresentam a modelagem interativa do sistema ciber-físico de uma comunidade inteligente, intitulado Net-in-Net. Como elucidado pelos autores, os sistemas ciber-físicos são ambientes que integram áreas físicas com a Internet, a fim de proporcionar interações de caráter eficiente, confidencial e oportuno entre aplicações e usuários. Tendo em vista que neste trabalho foram representados os fluxos de interação de um sistema complexo caracterizado por requisitos de segurança, estabilidade e confiabilidade, ele deveria ter sido formalizado com a utilização de uma linguagem robusta e verificável – uma escolha adequada para uma circunstância dessa magnitude. Todavia, a interatividade do Net-in-Net foi modelada utilizando-se simples fluxogramas, o que eventualmente resulta na criação de modelos ambíguos e subjetivos.

Mirlacher, Palanque e Bernhaupt (2012), apresentam uma abstração baseada em RdP para apoiar o projeto, implementação e verificação de *front-ends* que possuem animações interativas. A abordagem possui o diferencial de transformar os fluxos interativos em dois níveis diferentes de interação, nos quais atividades exclusivas são conduzidas. Enquanto a camada superior se concentra nos objetos interativos existentes, dando atenção a suas propriedades animadas e à composição dessas animações, o nível mais baixo de descrição abrange os aspectos temporais, preocupando-se com a utilização eficiente de *tweening* (interpolação) e possíveis problemas de conexão com o *hardware* responsável pela renderização das animações.

2.8.2 Modelagem formal de requisitos de software utilizando Redes de Petri

A fim de permitir que os requisitos de software sejam melhor definidos para que não haja retrabalhos, ou problemas no desenvolvimento de um software, existem diversas

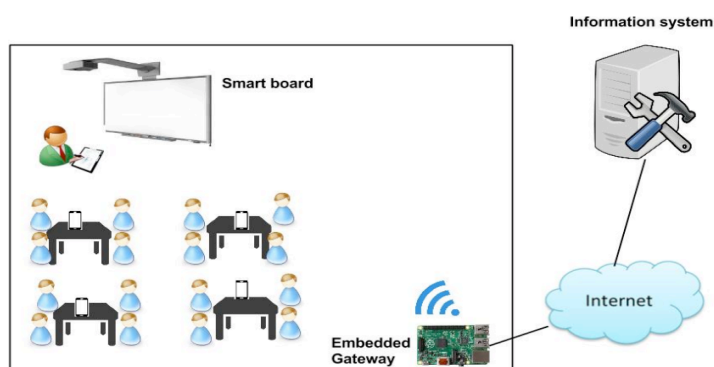
técnicas como Casos de Uso, *Viewpoints*, dentre outros, conforme levantamento feito por Fernandes (2012) em seu trabalho de pesquisa. Uma nova proposta para facilitar o processo de engenharia de requisitos é a utilização de Árvore de Requisitos proposta por Oliveira (2010), onde desenvolvedor e cliente trabalham juntos em um modelo gráfico para mútua compreensão do que se quer para o sistema.

Sobre os requisitos modelados em Árvore de Requisitos, Oliveira (2010) e Fernandes (2012), propõem a utilização de Redes de Petri Modal, que são Redes de Petri Coloridas com Expressões Lógicas (RdPeL) para verificação e validação dos requisitos.

2.8.3 Computação ubíqua

Bargaoui e Bdiwi (2014), expõem a importância da evolução do modelo de sala de aula tradicional, que se baseia na prática docente da figura do professor como responsável pelo saber. Para a utilização de um modelo mais ativo, ou seja, uma sala de aula mais colaborativa, faz-se necessário permitir que os alunos possam colaborar e compartilhar recursos através de novas tecnologias (*tablets, smartphones, quadros interativos, tabletops, etc.*). A computação ubíqua pode facilitar a aprendizagem colaborativa, criando um ambiente de aprendizagem inteligente. A sala de aula ubíqua deve ser capaz de suportar a interação de dispositivos heterogêneos conectados através de ligações sem fios a um *gateway*. Enquanto tecnologias como e-learning permitem ao aluno o contato com a internet, a tecnologia onipresente permite que os próprios dispositivos se conectem aos alunos.

A arquitetura proposta por Bargaoui e Bdiwi (2014), define que: a sala de aula *Ubiquitous* vista na Figura 7, possui três componentes principais: O primeiro componente é o sistema de informação: ele contém as ferramentas de administração e do servidor de banco de dados. Seus objetivos são o gerenciamento de usuários, gerenciamento remoto de várias salas de aula e gestão remota da segurança. O segundo componente é o gateway integrado, que permite o acesso remoto e troca de dados com dispositivos inteligentes. Finalmente, o terceiro componente são os dispositivos inteligentes e sensores. O estudante e o professor podem interagir com dispositivos como *smart board*, computador, sistema de resposta interativa, vídeo e dispositivos de áudio.

Figura 7 - *Ubiquitous Classroom*

Fonte: Bargaoui e Bdiwi (2014)

Richards, Petre e Bandara (2012), descreve um módulo, *My Digital Life*, criado para auxiliar estudantes novatos da graduação, da educação a distância, no desenvolvimento de programas ubíquos. Eles então experimentam, de maneira lúdica, a prática de conceitos de Computação Ubíqua através de um *hardware* chamado de SenseBoard, que pode ser programado por meio da linguagem *Sense*, uma linguagem em blocos semelhante ao *Scratch*⁴. O dispositivo SenseBoard é uma placa, com input, output e alguns sensores, baseado no microcontrolador Arduino⁵. O SenseBoard surgiu devido à necessidade de uma ferramenta mais simples do que Arduino ou *Phidgets*, por exemplo, para o ensino de computação ubíqua. A ideia era que não fossem exigidos muitos conhecimentos prévios, como em eletrônica, em circuitos ou em linguagem de programação. Isso então possibilitou o uso dessa ferramenta em aulas práticas de Computação Ubíqua para alunos novatos do curso de Ciência da Computação. Como o público-alvo era composto por alunos novatos, não se poderia exigir que estes já trouxessem uma bagagem de conhecimento em linguagens de programação. Por isso, foi adotada uma extensão à linguagem *Scratch*, o *Sense*, que se baseia na estrutura de ligação de blocos. É possível ver que (RICHARDS; PETRE; BANDARA, 2012), trabalha o ensino de conceitos de Computação Ubíqua por meio da experimentação prática desses conceitos, sem exigências de conhecimentos prévios em programação. No entanto, o ponto negativo é que, aparentemente, o trabalho foi descontinuado, pois o *hardware* SenseBoard não está mais disponível para a venda e a linguagem *Sense*, apesar de poder ser baixada, apresenta-se bastante instável, parando de funcionar com frequência.

⁴ <https://scratch.mit.edu/projects/editor/> 2<https://www.arduino.cc/en/Guide/Introduction>

⁵ <https://www.arduino.cc/en/Guide/Introduction>

Chin e Callaghan (2013), apresenta a ferramenta Programação Interativa Pervasiva (no inglês, *Pervasive Interactive Programming* - PIP), voltada para o ensino de habilidades elementares de programação em conjunto com conceitos de Internet das Coisas (no inglês, *Internet of Things* - IoT). O PIP é um cenário que envolve a programação de um conjunto de “coisas” para fornecer alguma funcionalidade coordenada (e.g. um reproduzidor de música para de tocar no momento em que um telefone começa a chamar). Assim, os estudantes aprendem conceitos básicos de lógica de programação (e.g. *if/else*) por meio da escrita de regras para a comunicação entre objetos, que representam “coisas” do mundo real. Como exemplo de uma prática com o PIP, tem-se que três componentes devem ser controlados: um telefone, a luz e um *media player*. Para isso, as regras devem ser escritas de tal forma que, quando o telefone chamar, as luzes aumentarão o brilho e o *media player* parará de tocar. O PIP poderia então ser usado para o ensino de alguns conceitos ubíquos a estudantes sem muita bagagem de programação. No entanto, ficaria mais limitado à prática da programação e da comunicação entre sensores e objetos, sem levar em consideração outros conceitos de Computação Pervasiva, como Adaptabilidade e Sensibilidade a Contexto, que não são possíveis de trabalhar neste ambiente sem aumentar consideravelmente a complexidade para o aluno.

Em Chalmers (2015), é apresentada a estrutura básica de um curso de Computação Ubíqua e, dentro dessa estrutura, estão as aulas práticas em laboratório, que duram em média duas horas. A teoria e a prática são executadas em paralelo e, mesmo que ligadas, ainda funcionam de maneira independente uma da outra. Dispositivos chamados *Phidgets* são usados nas aulas práticas, estes possuem *Application Programming Interfaces* (API) fornecidas em várias linguagens de programação. Por conta disso, 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. Isso permite que os alunos se concentrem melhor no tema proposto pelo projeto sem se preocupar com o aprendizado de alguma nova linguagem. O ponto negativo desse trabalho é o custo médio para aquisição dos dispositivos utilizados e a ausência de um ambiente de apoio às aulas práticas para auxiliar na visualização dos conceitos apresentados.

2.8.4 *Análise comparativa dos trabalhos relacionados*

Na análise referente aos trabalhos sobre especificação formal, Mirlacher, Palanque e Bernhaupt (2012) e Martinie, Navarre e Palanque (2014), são as abordagens que geram modelos com o maior nível de dificuldade de compreensão. Esse critério se justifica pelo fato dos mesmos trabalhos envolverem a maior quantidade de conhecimento obrigatório em áreas distintas a ser previamente compreendido pelos profissionais que objetivam utilizá-las. Em casos de especificação de sistemas complexos, como em Ma et al. (2010) e Ciardelli, Bixio e Regazzoni (2011), o uso de formalizações passíveis de verificação mostra-se como uma prática relevante, graças à possibilidade de se perceber e corrigir erros nas etapas iniciais de projetos.

Em relação aos trabalhos de Computação Ubíqua, percebe-se a utilização de kits *SenseBoard*, PIP e *Phidgets*. Não foram encontrados trabalhos que utilizem os *smartphones* como artefato para as aulas práticas de Computação Ubíqua, apesar desse dispositivo ser bem popular entre os estudantes.

Levando em consideração o trabalho de Hamdaoui, Alshammari e Guizani (2013) (2013), é possível afirmar que o sucesso e a popularidade de aplicações e serviços móveis resultaram em um aumento ainda mais significativo do número de dispositivos móveis no mundo inteiro. Esses dispositivos já se tornaram parte essencial na vida das pessoas, que os usam para os mais diversos fins e em diferentes contextos. Com base no exposto, torna-se então mais viável a utilização de *smartphones* para o apoio às aulas práticas de Computação Ubíqua. Isso devido à maioria dos estudantes possuir tal objeto, além do professor também poder dispor, com menor dificuldade de alguns aparelhos para a realização dessas práticas.

Nenhum dos trabalhos estudados está relacionado diretamente a perspectiva de se ter um ambiente físico de um laboratório sendo monitorado durante práticas nele efetuadas e, posteriormente, estes dados servirem não só para avaliação do professor mas também para simulação da prática por parte do aluno.

3 MODELAGEM FORMAL DO U-LABPA EM REDES DE PETRI COLORIDAS

Neste capítulo é apresentada a proposta deste trabalho tendo como contexto a arquitetura geral do *framework* para criação de atividades visando laboratórios que utilizam Computação Ubíqua. O trabalho realizado centra-se na modelagem por RdPC dos requisitos “Autenticação”, “Cadastra Atividade”, “Registra Atividade de Usuário”, “Disponibiliza Atividades de Usuário” e “Realiza Atividade” efetuando a validação destes, e observando se estes são suficientes para o sistema ou se existe a necessidade de acréscimo de novos requisitos. Através de simulações são verificados e validados os requisitos. Somente são trabalhados aspectos do *software* u-LabPA nas simulações, não sendo feitas tentativas de modelagem dos laboratórios reais, até por estarem estes fora do domínio de fenômenos discretos e, portanto, não estarem no escopo de modelagem das RdPC.

O u-LabPA (SARMENTO et. al., 2012), é um *framework* baseado em um conjunto de módulos de *software*, com suporte a componentes de *hardware* (e.g. WSN e RFID), que visa permitir que professores acompanhem atividades realizadas pelos alunos em laboratórios reais, assim como monitorem essas atividades em tempo real, durante sua execução. Este sistema é distribuído, tendo um servidor local de apoio em cada laboratório, para prevenir eventual perda de dados em caso de queda de conexão, e um servidor central na instituição de ensino a distância.

A motivação inicial para a modelagem do u-LabPA se deve à dificuldade em se instrumentalizar práticas laboratoriais no Ensino a Distância. Atualmente a EaD utiliza como principal meio de comunicação e operacionalização de seus cursos/disciplinas os Ambientes Virtuais de Aprendizagem como plataforma tecnológica de comunicação. Porém, tais ambientes não contemplam ações de monitoramento e análise de práticas laboratoriais pelo professor.

Os módulos de software presentes no u-LabPA são:

- Ambiente Laboratorial de Monitoramento de Atividades (ALMA): centraliza o cadastro de informações referentes a usuários, perfis, laboratórios, atividades, e interage com os demais sistemas relacionados ao u-LabPA;
- Sistema de Monitoramento e Identificação em Laboratório Real (SMIL): responsável pela identificação, localização e sensoriamento dos indivíduos e objetos presentes no laboratório, diretamente relacionados às atividades a serem realizadas;
- Sistema de Localização de Objetos e Pessoas (SLOP): sistema auxiliar ao SMIL que

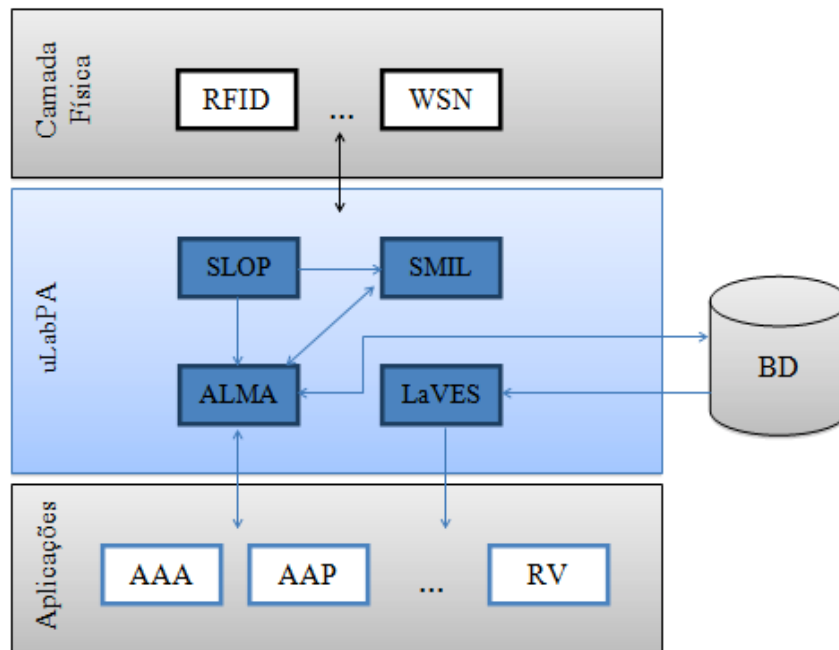
permite a localização de pessoas e equipamentos - este subsistema tem direta relação com as WSN e leitores RFID, por exemplo;

- Laboratório Virtual para Experiências Simuladas (LaVES): sistema responsável pelo provimento dos dados para simulação em laboratórios 3D utilizando técnicas de Realidade Virtual.

O SLOP é responsável pela descoberta de objetos e pessoas no laboratório, enquanto o SMIL é responsável pela agregação (relacionamento) de objetos e pessoas no laboratório, assim como pela detecção de eventos (interpretação) relacionados aos *widgets*⁶ (sensores e atuadores).

O *framework* u-LabPA foi criado para ser utilizado como parte de aplicações que permitam a criação de atividades, e intermediar a comunicação destas aplicações com uma Camada Física (e.g. sensores e atuadores) e de consistência (base de dados - BD). Esta estrutura pode ser vista na figura abaixo.

Figura 8 -Visão geral do sistema u-LabPA



Fonte: Sarmento (2016)

⁶ O termo *widget* é uma abstração para sensores e atuadores que estejam disponíveis no ambiente. Normalmente é usado para sensores, segundo o modelo de Dey e Abowd (2000).

Na Figura 8 são consideradas as aplicações AAA, AAP e RV. Estas são respectivamente o Ambiente de Atividades Laboratoriais do Aluno, Ambiente de Atividades Laboratoriais do Professor e aplicação Realidade Virtual. As duas primeiras permitem a interação com formulários de coleta de dados das atividades laboratoriais, bem como dados complementares coletados pelos sensores presentes no laboratório; além da criação e avaliação destas atividades. A terceira permite colher as informações do laboratório descritas pelo aluno, pelas informações sensoriais complementares e pelo contexto, e simular a atividade realizada pelo aluno em ambiente de Realidade Virtual. Desta forma, pode-se notar que o *framework* não é uma aplicação e sim uma estrutura básica para construir diferentes aplicações que permitam criar, acessar e manipular as atividades laboratoriais, haja vista que as aplicações envolvidas podem existir separadamente, ou em um AVA. Além disso, o tipo de laboratório também não é fixado para o u-LabPA, ele é uma abstração, podendo ser adaptado para diferentes contextos de laboratório.

Para que se possa ter uma modelagem do fluxo “Realização de Atividades do *framework*”, foram utilizados não só os elementos de requisito do *framework*, mas também as aplicações AAP e AAA, que contém parte dos elementos de requisitos do u-LabPA. Isto se deve ao fato que o *framework* em si serve como arcabouço para se construir as aplicações que irão ser usadas para a criação e execução das atividades. A AAP e AAA são os programas exemplos que são usados por parte do professor e do aluno para acesso às atividades. Assim, através deles, as funções do *framework* podem ser melhor percebidas.

Alguns dos requisitos utilizados podem ser vistos na Figura 09. Os demais se encontram no Anexo A.

Estes requisitos foram fornecidos pelo autor do u-LabPA. O desafio é mostrar que a modelagem formal poder ajudar a encontrar novos requisitos não percebidos, verificar os existentes, validá-los e, finalmente, permitir confrontar a modelagem UML feita por Sarmiento (2016), com a análise dinâmica proveniente de RdPC.

Figura 9 - Casos de Uso de Autenticação e Realiza atividade

RF001 Autenticação

Descrição	São recebidos valores para autenticação do usuário, login e senha ou crachá e senha, e são devolvidas as permissões, caso existam, ou null, caso não haja autenticação.
Fluxo principal	<p>(a) Framework recebe dados para autenticação de usuário;</p> <p>(b) Verifica se autenticação pode ser feita;</p> <p>(c) Em caso positivo, retorna conjunto de permissões do usuário;</p> <p>(d) Caso contrário, retorna null.</p>

RF002 Realiza atividade

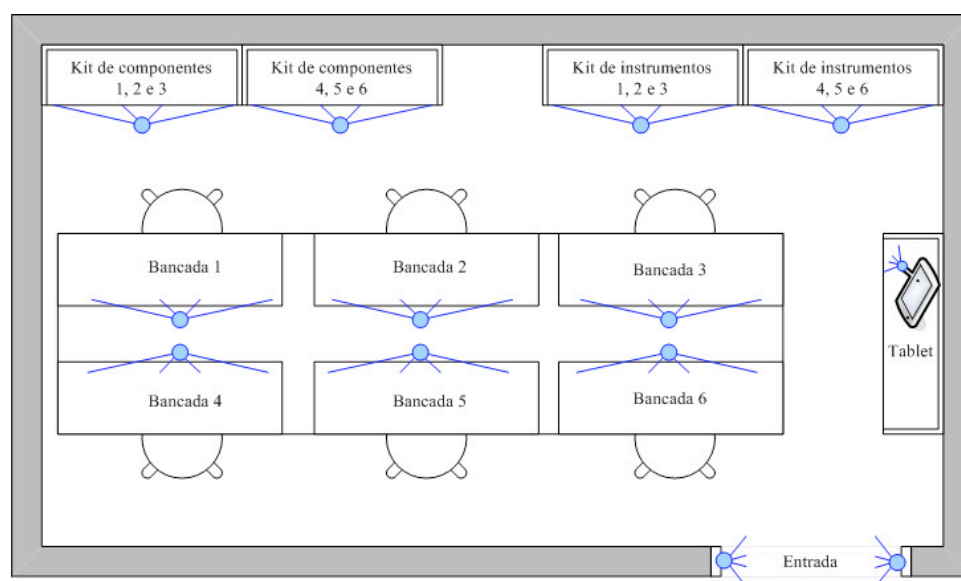
Descrição	São realizadas as práticas propostas pelo professor para o aluno que estiver em um laboratório. Pode constar de uma ou mais práticas.
Fluxo principal	<p>(a) Aluno abre formulário de Atividades;</p> <p>(b) Aluno acessa componentes, equipamento e bancadas para realização de prática (todo esse processo deverá ser monitorado pelo sistema);</p> <p>(c) Aluno anota os valores pedidos no formulário de Atividades;</p> <p>(d) Dados dos alunos e dados sensorizados de apoio são enviados para o servidor.</p>
Exceções	<ul style="list-style-type: none"> Falha no envio de dados. Nesse caso, envia alternativamente os dados para o servidor local do laboratório para atualizar o servidor central quando for reestabelecida a conexão (NS002).

Fonte: Sarmento (2016)

3.1 Cenário proposto

A fim de permitir a realização da modelagem formal e da simulação foi criado um cenário de um Laboratório Genérico com um conjunto de 06 bancadas, 27 alunos, 3 professores a distância, sensores e etiquetas de identificação RFID, conforme pode ser visto na Figura 10.

Figura 10 - Laboratório Modelado



Fonte: elaborado pelo autor.

No estudo de caso proposto, as atividades práticas em laboratório são definidas pelo professor por meio do AAP. Os alunos têm acesso a estas atividades através de uma ferramenta desenvolvida para seu perfil que permite visualizar o problema proposto pelo professor, iniciar a prática e indicar se os passos da atividade foram ou não realizados por meio do AAA.

A fim de permitir a identificação das pessoas presentes no ambiente do laboratório real, estes indivíduos portam crachás com as etiquetas de Rádio Frequência (RF). Nestes locais, o acesso ao sistema de atividades é feito através de dispositivos móveis como *tablet* ou *smartphone*.

As variáveis físicas como tensão, corrente e resistência elétrica, por exemplo, são monitoradas por sensores da rede *Wireless Sensor Network* (WSN).

Os experimentos realizados pelos usuários presentes no laboratório são executados sobre áreas pré-definidas da bancada a fim de que se possa obter o relacionamento pessoa-objeto-experimento, necessário ao monitoramento e armazenamento das atividades dos participantes. Estes dados são vinculados pelo sistema à atividade do aluno, posicionado ou inseridos em um Laboratório Virtual para simulação da prática, podendo ser acessados por um professor remotamente.

Este é o cenário sobre o qual podem ser criadas a AAP e AAA, bem como os servidores local e a distância do u-LabPA para que o sistema possa funcionar. É sobre este cenário que o *software* do u-LabPA deve funcionar, portanto, seus requisitos devem ser verificados e validados. A modelagem foi feita para verificação destes requisitos.

3.2 u-LabPA – fluxo de atividades

No u-LabPA o aluno começa a ser monitorado quando entra no laboratório, onde deve passar um cartão de identificação para a liberação do seu acesso. Depois de autenticado, ele solicita uma atividade ao sistema. Em seguida, o sistema verifica a disponibilidade em seu banco de dados, previamente alimentado pelos professores e, aleatoriamente, disponibiliza a atividade ao aluno. Após receber sua atividade, o aluno solicita ao sistema uma bancada e os materiais a serem usados na execução desta atividade. Já acomodado em uma bancada e com seus materiais disponíveis, o aluno desenvolve a atividade, informando todos os dados lidos no experimento. Na sequência, estes dados são armazenados em um banco de dados para consultas posteriores. Finalizando a atividade, o aluno libera a bancada e devolve os instrumentos e componentes, podendo agora passar o crachá e deixar o laboratório. O aluno não é obrigado a seguir o fluxo como foi descrito acima. Ao entrar, se ele decidir sair sem ao menos ter selecionado uma atividade, ele pode deixar o laboratório. Após ter uma atividade selecionada, o aluno pode resolver sair sem completá-la e posteriormente retomá-la. Nesta situação, pode também solicitar uma nova atividade ao sistema, tendo total liberdade para ter êxito em sua atividade.

3.3 Cenário geral para simulação com RdPC

3.3.1 *Seguem a descrição de cada um dos modelos criados para o fluxo “Realização de Atividades do u-LabPA”.*

3.3.1.1 *Descrição do cenário:*

O cenário geral definido para o funcionamento do u-LabPA (ver Figura 11) prevê a disponibilização de interfaces de acesso para professores e alunos. O trabalho de Sarmiento et al. (2012), apresenta um detalhamento do fluxo de acesso ao ALMA. Este trabalho abrangeu em sua modelagem oito cenários. A utilização de RdPC permite que, além de uma especificação formal do cenário, tenha-se uma visão geral do sistema. Tal característica permitiu a melhoria do modelo proposto por Sarmiento, sendo otimizado para cinco cenários, são eles: 01 – Autenticação, 02 – Alocação de Atividades, 03 – Alocação de Instrumentos, Componentes e Bancadas, 04 – LaVES e 05 – Devolução de Instrumentos, Componentes e Bancadas (PEQUENO FILHO et al., 2016). A modelagem destes cenários permitiu uma melhor visualização da proposta do framework e na identificação do fluxo de Realização de Atividades por parte dos alunos. Esse fluxo constitui uma parte central para estudo do comportamento do u-LabPA por envolver dois dos principais atores do sistema – professor e aluno – e atividades que perpassam os diversos fluxos do sistema, como pode ser visto na Figura 2. O cenário geral do fluxo trata da entrada do aluno no laboratório, seu login, a seleção de uma atividade pelo professor, e sua disponibilização aos alunos. Assim, quando o aluno entra no laboratório e tem uma atividade ativa, ele pode solicitar a realização desta. Depois ele solicita instrumentos, componentes e local da bancada para realizar sua atividade. Por fim devolve o material solicitado e realiza o logout no sistema. O aluno pode, ainda, simular a atividade realizada em um ambiente virtual.

das transições pode ser disparada por vez;

Opção 1: O usuário opta por deixar a sala, situação representada pela transição “Usuario decide deixar a sala”. Caso esta seja disparada, a transição de substituição “Usuario pega inst., Comp. e Solicita Banc.” é desabilitada para este usuário e uma ficha é depositada no lugar “ALMA Autenticacao”, sendo possível seu retorno ao laboratório em um momento posterior, pois, efetuando novamente o *login* no sistema, esta tarefa abandonada por ele estará disponível para ser retomada;

Opção 2: Se o usuário decidir realizar sua atividade, situação representada pela transição de substituição “Usuario pega inst., Comp. e Solicita Banc.”, esta é disparada, sendo a transição “Usuario decide deixar a sala” desabilitada e uma ficha é depositada no lugar “Usuario com inst. Comp. e Banc.”.

Ação 4. O usuário está pronto para realizar sua atividade, situação representada pelo lugar “Usuario com inst. Comp. e Banc.”. Neste caso, a transição de substituição “Usuário realizando a Atividade” é disparada e uma ficha é depositada no lugar “Usuario finaliza a Atividade”.

Ação 5. O usuário concluiu a atividade, situação representada pelo lugar “Usuario finaliza a Atividade”. Neste caso, a transição de substituição “Usuario devolve comp., inst. e banc.” é habilitada, e após seu disparo uma ficha é depositada no lugar “Usuario pronto para deixar a sala”.

Ação 6. O usuário conclui sua atividade, devolve os instrumentos e componentes usados para a realização da atividade e a bancada usada por ele é liberada. Assim, ele está pronto para efetuar *logout*, situação representada por uma ficha no lugar “Usuario pronto para deixar a sala”. Neste caso, a transição “Aluno efetua LogOut” é habilitada e seu disparo remove a ficha do lugar “Usuario pronto para deixar a sala”.

Para a modelagem geral do sistema, foram encontradas dificuldades na elicitação de requisitos devidos aos componentes dos módulos ALMA, SLOP e SMIL não estarem completamente descritos por Sarmiento (2016). Para os módulos ALMA (responsável pela centralização do cadastro de informações), SLOP (que tem atribuição de ser a camada de abstração de software sobre o *hardware* dos sensores) e SMIL (responsável pelo sensoriamento do laboratório físico), as informações foram obtidas através de entrevistas com o cliente⁷, de forma a preencher as lacunas encontradas no processo de modelagem da RdPC geral do sistema.

⁷ Para efeitos de simulação, o autor do *framework* foi considerado cliente neste trabalho.

O fluxo da rede foi mostrado ao cliente de uma forma que ele pudesse saber o que estava se passando em cada etapa do sistema. O método utilizado neste e nos demais cenários foi o de Protótipos Evolucionários⁸ a fim de se garantir que realmente pudesse ter uma especificação mais atualizada do sistema. A Prototipação Evolucionária parte de um sistema inicial que é exposto ao cliente e então comentado, passando por estágios de aperfeiçoamento, até que se chegue a um produto adequado aos interesses deste cliente (SOMMERVILLE, 2005). As RdPC, neste caso, foram usadas como protótipos, passando por processo de melhorias contínuas até seu estágio final.

No entanto, isto só foi possível pela peculiaridade do cliente também conhecer RdPC. Para outro perfil de cliente, é provável que os protótipos criados não ficassem tão claros. Neste caso, outras técnicas intermediárias entre as RdPC e os requisitos podem ser utilizadas a fim de facilitar a comunicação entre o analista de requisitos e o cliente, tal como o uso de Protótipos em Papel ou protótipos em *Mockup*⁹.

No entanto, já se percebe uma maior semelhança entre o que foi modelado em RdPC e o Diagrama de Atividade do u-LabPA visto na Figura 02, onde vê-se o fluxo atual das atividades do sistema.

3.3.2 Modelo “Autenticação e Cadastra Atividade”

3.3.2.1 Descrição do cenário:

A RdPC analisada nesta seção pode ser vista na Figura 13 (“Modelo RdPC de Autenticação e Cadastra Atividade”). Ela está ligada aos requisitos apresentados na Figura 12 (“Autenticação e Cadastra da atividade”) e é a transição de substituição “Usuario passa o cartão para entrar” da Rede de Petri Colorida mostrada na Figura 11.

Todo o requisito “Realização de Atividades” é feito pelo aluno, porém, “Disponibiliza Atividades” é do perfil professor. Não seria possível o primeiro requisito sem o segundo. Apesar de elementar, essa situação só pôde ser verificada após as simulações, quando se percebeu que havia uma inconsistência no diagrama de atividades original.

⁸ É modelo de desenvolvimento cujo objetivo é trabalhar com os clientes e evoluir um sistema final a partir de uma especificação genérica inicial (SOMMERVILLE, 2011)

⁹ Apresentação das funcionalidades em forma gráfica, contendo os elementos de interação (botões, menus etc.), arquitetura da informação, disposição dos componentes gráficos e textuais, tipo de fonte e cores a serem usadas.

Figura 12 - Requisitos: “Autenticação” e “Cadastra Atividade”

RF001 Autenticação

Descrição	São recebidos valores para autenticação do usuário, login e senha ou crachá e senha, e são devolvidas as permissões, caso existam, ou null, caso não haja autenticação.
Prioridade	1
Fluxo principal	(a) Framework recebe dados para autenticação de usuário; (b) Verifica se autenticação pode ser feita; (c) Em caso positivo, retorna conjunto de permissões do usuário; (d) Caso contrário, retorna null.

RF006 Cadastra atividade

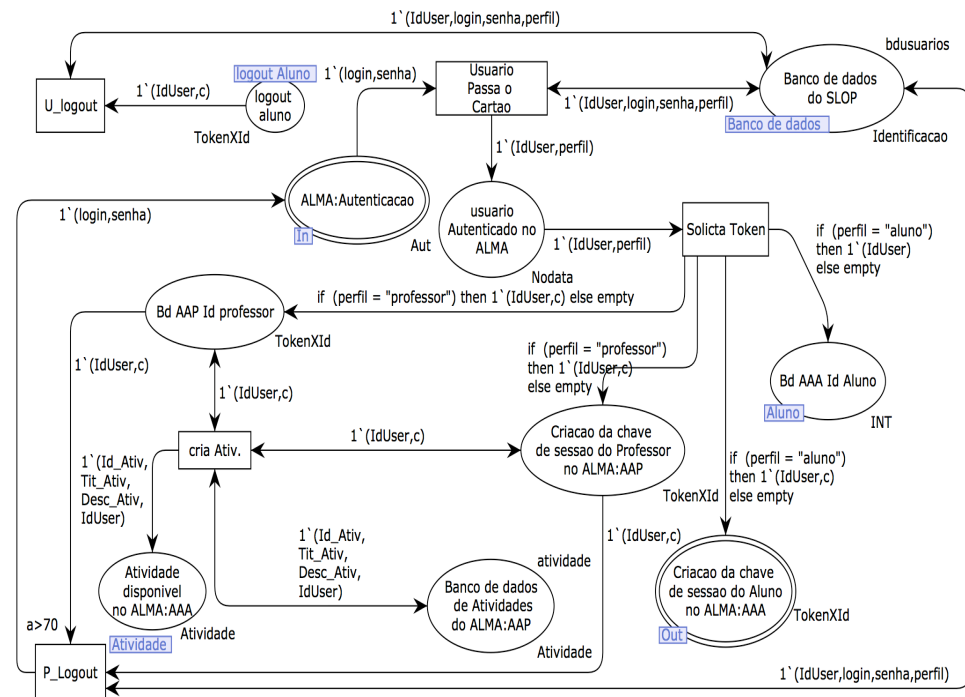
Descrição	São recebidos os dados de atividade e é realizado o cadastro, retornando erro caso haja falha.
Prioridade	2
Fluxo principal	(a) Framework recebe dados necessários para cadastrar uma nova atividade; (b) Caso o cadastro seja feito com sucesso, retorna ID da nova atividade.
Fluxo alternativo	(a) Framework recebe dados para alterar atividade.

Fonte: Sarmento (2016)

3.3.2.2 Descrição da RdPC:

Na Figura 13 pode-se observar o modelo RdPC de autenticação e criação de atividades, situação representada pela transição de substituição “Usuario passa o cartao para entrar”, Figura 11. Nesta subrede, o fluxo desempenhado é apresentado da seguinte forma:

Figura 13 - Modelo RdPC de Autenticação e Cadastra Atividade



Fonte: elaborado pelo autor.

Ação 1. A transição “Usuario Passa o Cartao” só está habilitada se houver uma ficha no lugar “Alma:Autenticacao” e a mesma constar no lugar “Banco de dados do SLOP”. Esta condição sendo atendida, a transição é disparada e o usuário ingressa no ambiente, situação representada pelo lugar “usuario Autenticado no ALMA”.

Ação 2. Com uma ficha no lugar “usuario Autenticado no ALMA”, a transição “Solicita Token” é habilitada e seu disparo deposita fichas nos lugares:

Opção 1 - Aluno: “Bd AAA Id Aluno” e “Criacao da chave de sessão do aluno no ALMA:AAA”. O lugar “Bd AAA Id Aluno” representa o banco de dados que o identifica no sistema. O lugar “Criacao da chave de sessão do aluno no ALMA:AAA” representa a identificação segura (criptografia) recebida pelo aluno no sistema, a fim de promover segurança nos dados trafegados. Se o usuário autenticado no sistema for um aluno, esta condição é checada no arco que liga a transição “Solicita Token” ao lugar “Bd AAA Id Aluno” e “Criacao da chave de sessão do aluno no ALMA:AAA”. Caso o sistema identifique que o perfil é de um aluno, o sistema libera o acesso, depositando fichas nos lugares “Bd AAA Id Aluno” e “Criacao da chave de sessão do aluno no ALMA:AAA”.

Opção 2 - Professor: “Bd AAP Id professor” e “Criacao da chave de sessao do professor no ALMA:AAP”. O lugar “Bd AAP Id professor” representa o banco de dados que

o identifica no sistema. O lugar “Criação da chave de sessão do professor no ALMA:AAP” representa a identificação segura (criptografia) recebida pelo professor no sistema, a fim de promover segurança nos dados trafegados. Estes lugares recebem uma ficha cada, somente se o usuário autenticado no sistema for um professor. Esta condição é checada no arco que liga a transição “Solicita Token” ao lugar “Bd AAP Id professor” e “Criação da chave de sessão do professor no ALMA:AAP”. Caso o sistema identifique que o usuário possui o perfil “professor”, o sistema libera o acesso, depositando ficha nos lugares “Bd AAP Id professor” e “Criação da chave de sessão do professor no ALMA:AAP”.

Ação 3. Com uma ficha nos lugares: “Bd AAP Id professor”, “Banco de dados de Atividades do ALMA:AAP” e “Criação da chave de sessão do professor no ALMA:AAP”, a transição “cria Ativ.” é habilitada. O sistema é composto de um banco de dados de atividades representado pelo lugar “Banco de dados de Atividades do ALMA:AAP”, onde o professor pode disponibilizar quantas atividades forem necessárias para sua(s) turma(s). Ressaltando que cada aluno faz uma atividade, então, para uma turma de 30 alunos, por exemplo, o professor deve disponibilizar 30 atividades. A transição “cria Ativ.” é disparada e coloca uma ficha no lugar “Atividade disponível no ALMA:AAA”.

Ação 4. A transição “P_Logout” é habilitada quando os lugares: “Bd AAP Id professor”, “Criação da chave de sessão do professor no ALMA:AAP” e o “Banco de dados do SLOP” tiverem uma ficha cada do mesmo usuário. Seu disparo indica que o professor autenticado no sistema decidiu sair do ambiente.

3.3.3 Modelo “Visualiza e Abre Atividade”

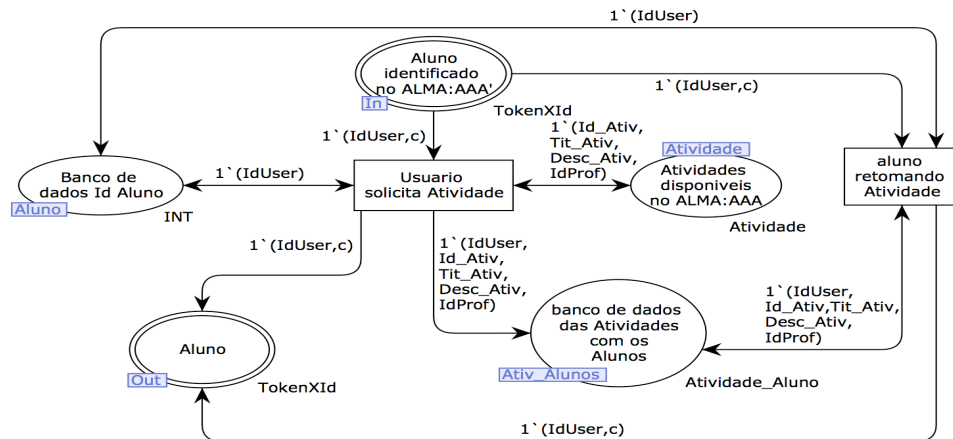
Esta parte do requisito “Realiza Atividade” está dividida em duas redes devido a sua complexidade. Estas RdPC são: 1. Alocação de Atividades; 2. Alocação de instrumentos, componentes e bancadas. Abaixo segue a primeira das duas redes que abrangem este requisito.

3.3.3.1 Descrição do cenário:

Na modelagem apresentada na Figura 14 o aluno solicita uma atividade previamente disponibilizada, sendo esta alocada a ele pelo sistema de forma aleatória - em um Lugar com **n** Fichas é sorteado pelo *software* CPNTools uma de cada vez para ser a atividade -, conforme a disponibilidade ofertada pelo professor. Se o aluno já entrou anteriormente no

sistema e resolveu sair antes de concluir sua atividade, pode concluí-la ou solicitar uma nova atividade ao sistema. Isso é representado pela transição “aluno retomando Atividade”, na Figura 14.

Figura 14 - Modelo RdPC de Alocação de Atividades



Fonte: elaborado pelo autor.

3.3.3.2 Descrição da RdPC:

Na Figura 14 pode-se observar o modelo RdPC de alocação de atividades, operação representada pela transição de substituição “Usuario Solicita a Atividade no Alma:AAA” na Figura 11. Nesta subrede, o fluxo desempenhado é apresentado da seguinte forma:

Ação 1. A transição “Usuario solicita Atividade” só é habilitada se o aluno está identificado no sistema, situação representada por uma ficha no lugar “Aluno identificado no ALMA:AAA”, e se o professor disponibilizou alguma atividade no ambiente, situação representada por uma ficha no lugar “Atividades disponiveis no ALMA:AAA”. Satisfeitas as condições, a transição está habilitada e pode ser disparada. Disparada a transição, os lugares “banco de dados das Atividades com os Alunos” e “Aluno” recebem uma ficha cada, informando ao sistema que determinado aluno está com uma atividade.

Ação 2. A transição “aluno retomando Atividade” só está habilitada para o(s) aluno(s) que já fizeram o *login* no ambiente anteriormente, representado pela transição “Usuário solicita Atividade” e, em seguida, resolveram deixar o ambiente sem realizar a atividade, representado pela transição “Usuario decide deixar a sala” que pode ser vista na Figura 11. Para estes alunos, o ambiente dá a opção de escolherem entre continuar a atividade abandonada (“aluno retomando Atividade”) ou iniciar uma nova atividade (“Usuario solicita

Atividade”). Assim, as duas transições estão habilitadas, porém, somente um disparo é permitido, tendo o aluno que escolher entre elas.

Opção 1: Optando o aluno por dar continuidade à atividade alocada anteriormente, a transição “aluno retomando Atividade” é disparada. Neste caso, a transição “Usuario solicita Atividade” é desabilitada para este aluno e uma ficha é depositada no lugar “Aluno” contendo dados de identificação segura (criptografia) deste aluno. A transição “Usuario solicita Atividade” pode voltar a ficar habilitada caso o aluno resolva novamente abandonar a atividade e retornar ao ambiente posteriormente, onde este processo se repete.

Opção 2: A ação 1 é repetida.

3.3.4 Modelo alocação de instrumentos, componentes e bancada para usuário/realização e anotação da atividade

Esta parte do requisito “Realiza Atividade” trata do monitoramento do aluno dentro do laboratório, bem como do processo de realização da atividade propriamente dita. Ela também aborda a parte de Contexto de Laboratório. Após a atividade feita, ela é registrada no sistema pelo aluno.

3.3.4.1 Descrição do cenário:

Na Figura 15, foi modelada a alocação de instrumentos, componentes e bancadas. Neste caso, o aluno solicita uma bancada e pega os instrumentos e componentes necessários para a realização da atividade a ele alocada. Havendo disponibilidade, o sistema registra a identificação deste usuário, a bancada, os instrumentos e os componentes utilizados.

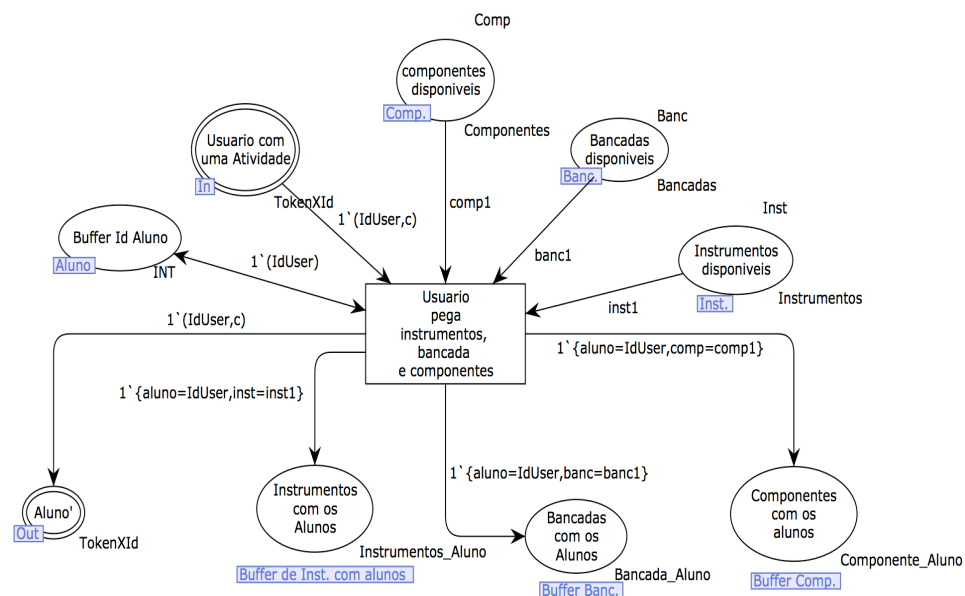
3.3.4.2 Descrição da RdPC:

Na Figura 15 pode ser observado o modelo RdPC para alocação de instrumentos, componentes e bancada para usuário.

Ação 1. A transição “Usuario pega instrumentos, bancada e componentes” está habilitada quando o usuário está com sua atividade alocada (representada por uma ficha no lugar “Usuario com uma Atividade”), quando uma bancada está disponível (representada por uma ficha no lugar “Bancadas disponíveis”), existem componentes disponíveis (representada por uma ficha no lugar “componentes disponíveis”) e, por fim, possui instrumentos

disponíveis (representada por fichas no lugar “Instrumentos disponíveis”). Se algum destes lugares não possuir fichas deste usuário, a transição não está habilitada. Se estes requisitos forem atendidos, a transição está habilitada para ser disparada e seu disparo deposita fichas no repositório de Instrumentos (lugar “Instrumentos com os Alunos”), Bancadas (lugar “Bancadas com os Alunos”) e componentes (lugar “Componentes com os Alunos”), colocando também uma ficha no lugar “Aluno”.

Figura 15 - Modelo RdPC para Alocação de Instrumentos, Componentes e Bancada para usuário



Fonte: elaborado pelo autor.

3.3.5 Criação de uma instância virtual do laboratório

Os dados foram colhidos do sistema (Figura 11) e podem ser usados em aplicações, como uma aplicação de Realidade Virtual para Simulação do Laboratório. Note que, para verificar os dados requisitados do *framework*, foi necessário simular o ambiente onde os dados gerados por este seriam usados.

3.3.5.1 Descrição do cenário:

Na Figura 16 pode se observar a rede de Petri colorida que descreve o comportamento do laboratório virtual para experiências simuladas, denominado LaVES. Nesta aplicação, a interface de dados do *framework* homônima pode entregar as informações

Ação 2. Com uma ficha no lugar “Bd Id Aluno”, a transição “Entrada dados praticos” é habilitada e seu disparo coloca uma ficha com os dados do experimento deste aluno no lugar “XML_In”.

Ação 3. Com uma ficha no lugar “XML_In”, “Bd Info Alunos/Ativ.” e “Banco de dados Id Aluno”, a transição “Entrada dados relac. a ativ. no LAVES” fica habilitada. Disparando esta transição, uma ficha mesclando os dados das fichas dos lugares “XML_In” e “Bd Info Alunos/Ativ.”, juntamente com o Id do aluno (lugar “Banco de dados Id Aluno”), será colocada no lugar “entrada de XML”.

Ação 4. Com uma ficha no lugar “entrada de XML”, a transição “carregar cenario” é habilitada, seu disparo coloca uma ficha no lugar “objetos carregados”, acrescentando à ficha as informações dos objetos necessários à construção da simulação da prática realizada.

Ação 5. Com uma ficha no lugar “objetos carregados”, a transição “execucao” é habilitada para disparo. Realizando o disparo desta, uma ficha é depositada no lugar “simulacao realizada”.

Ação 6. Com uma ficha no lugar “simulacao realizada”, a transição “finalizar” é habilitada e seu disparo coloca uma ficha no lugar “Usuario finaliza a tarefa”.

3.3.6 Devolução de instrumentos, componentes e bancada para usuário

Na Figura 18, o usuário finalizou sua atividade e se prepara para deixar o laboratório. Tal prática encerra o requisito Realiza a Atividade, liberando os equipamentos para outro usuário.

3.3.6.1 Descrição do cenário:

Na Figura 18 pode ser observado o modelo que simula a saída do aluno da sala. Após finalizar sua atividade, ele se prepara para deixar a sala. Se o aluno tem algo alocado, deve devolvê-lo primeiro, caso contrário, ao tentar deixar a sala o sistema informa que existem pendências a serem resolvidas, solicitando que o aluno as corrija antes de deixar a sala.

3.3.6.2 Descrição da RdPC:

Na Figura 18 é apresentado o modelo RdPC que representa a devolução de Instrumentos, Componentes e a liberação da Bancada pelo usuário (correspondendo à transição de substituição “Usuario devolve comp., inst. e banc.” visto na Figura 11). Nesta subrede, o fluxo desempenhado é apresentado a seguinte forma:

Ação 1. A transição “Usuário Devolve comp. Banc. e Inst.” é habilitada quando o lugar “Usuario finaliza a atividade” (Figura 11) contiver uma ficha. Na Figura 18, o usuário é representado por uma ficha no lugar “Usuário com a Atividade finalizada”.

Ação 2. O disparo da transição “Usuário Devolve comp. Banc. e Inst.” coloca uma ficha nos lugares: “Instrumentos devolvidos”, “Bancadas disponiveis”, “componentes devolvidos” e, por fim, “Aluno” contendo a criptografia dos dados de identificação deste aluno.

3.3.7 Registra atividade de usuário

Nas redes “Autenticação” (Figura 13) e “Alocação de Instrumento, componentes e bancada” (Figura 15) para usuário, percebem-se quatro lugares chamados “BD AAA ID Aluno”, “Instrumentos com os alunos”, “Componentes com os alunos” e “Bancadas com os Alunos”. Este são lugares que agregam informações de contexto do aluno que está fazendo a prática dentro do sistema. Desta forma, o sistema pode executar o requisito Registra Atividade do Usuário como visto na Figura 17.

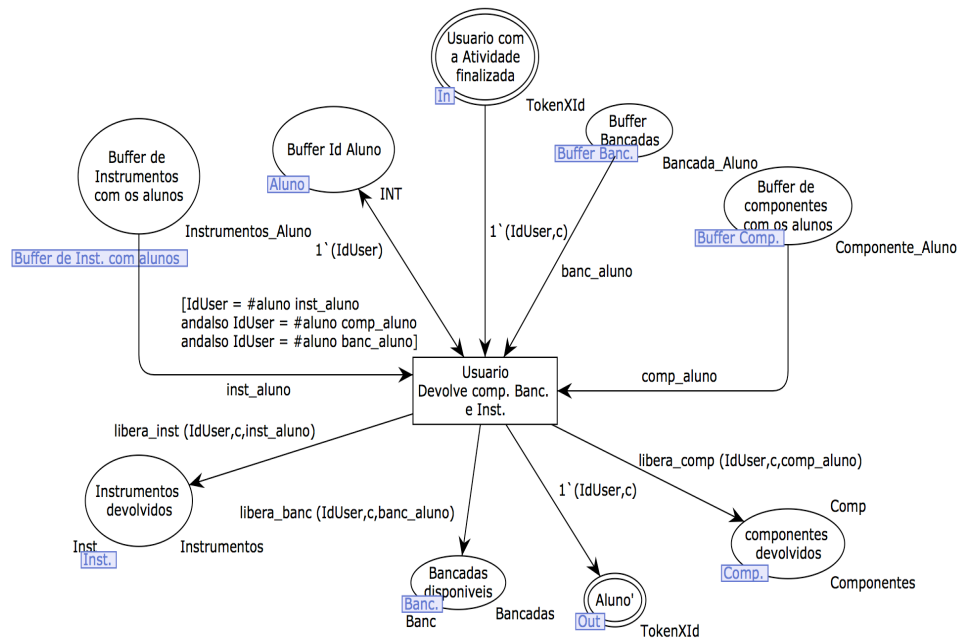
Figura 17 - Requisito - Registra atividade de usuário

RF007 Registra atividade de usuário

Descrição	Recebe para armazenamento dados de usuário, local, data e contexto atual da atividade sendo realizada.
Prioridade	1
Fluxo principal	(a) Recebe usuário, atividade e o que foi feito - registra individualmente cada ação do usuário relacionada a uma atividade, venha ela de uma leitura de <i>widget</i> ou de uma entrada manual.

Fonte: Sarmento (2016)

Figura 18 - Modelo RPC da Devolução de Instrumentos, Componentes e Bancada para Usuário



Fonte: elaborado pelo autor.

As RdPC apresentadas neste capítulo cobrem os requisitos de “Autenticação”, “Cadastra Atividade”, “Registra Atividade de Usuário”, “Disponibiliza Atividades de Usuário” e “Realiza Atividade” para verificação e validação dos mesmos, bem como sua interligação para a formação de um fluxo chamado Realização de Atividades. Este processo envolveu a fase de modelagem e simulação do sistema, porém, sem envolver diretamente informações das iterações feitas em cada etapa. Esta elaboração visa avaliar como o sistema se comportaria com um dado volume de 30 usuários fazendo uso do sistema, em diversas formas. Analisar este comportamento em relação à média de uso, máxima de uso, mínima de uso, permite ter uma visão mais clara do comportamento do sistema em termos de seus pontos de estresse, de possíveis *livelock*¹⁰, *deadlock*¹¹, finalização apropriada do sistema, finalização inesperada do sistema e condições de erro no fluxo do sistema após um número grande ou aleatório de iterações.

No Capítulo 4 é apresentada a análise dos modelos RdPC desenvolvidos para validação do sistema.

¹⁰ Um conjunto de processos ficariam executando infinitamente (estado ativo), onde cada processo esperaria o outro para que pudesse encerrar sua atividade (Tanenbaum, 2008).

¹¹ Um conjunto de processos está em *deadlock* se cada processo estiver esperando por um evento que somente outro processo do conjunto puder causar, estando todos em estado de Espera (Tanenbaum, 2008).

4 ANÁLISE DE RESULTADOS

A modelagem dos requisitos do *framework u-LabPA* foi feita por meio de cenários, utilizando as Redes de Petri Coloridas Hierárquicas, através da ferramenta CPN Tools¹² versão 4.0. O computador utilizado para as simulações foi um MacBook Pro com 16 GB de Memória RAM DDR3 e Microprocessador Intel core i5.

O principal intuito da modelagem e simulações foi verificar possíveis problemas que poderiam ocorrer em alguns dos requisitos especificados para o *framework u-LabPA*, envolvendo o fluxo de Realização de Atividades.

No modelo RdPC que representa o fluxo de Realização de Atividades do u-LabPA, que pode ser visto na Figura 11, foram utilizados Monitores. A análise dos dados monitorados é apresentada neste capítulo, assim como suas respectivas validações.

A simulação foi realizada para um cenário contendo 30 usuários – número médio de uma turma semi-presencial da UFC Virtual¹³, sendo 3 professores (um professor e dois professores tutores) e 27 alunos. Foram monitorados os passos dos alunos durante a realização das atividades de laboratório. Definiu-se a restrição de haver apenas uma atividade concluída por usuário a fim facilitar o monitoramento de seu comportamento nas redes e identificação de problemas. Foi escolhida esta condição para que se pudesse ter certeza do fluxo de dados dentro dos modelos de RdPC e o processo de depuração destes dados (e.g. fichas que eram esperadas em um lugar não chegaram ou transições não foram ativadas na rede).

Ao todo foram efetuadas 100 simulações, observando-se progressivamente os resultados para 10, 50 e 100 simulações, não sendo registradas flutuações significativas. Foi utilizado o máximo de 10.000 (dez mil) passos por simulação¹⁴, sendo monitorados 10 usuários escolhidos de forma aleatória. O número de passos foi escolhido de forma heurística, sendo suficientemente alto para permitir que todos os usuários pudessem concluir as atividades em uma simulação. Foram colhidos resultados de simulação referentes às seguintes atividades: (i) entrada de usuários no sistema do laboratório; (ii) usuários que saíram antes de concluir as atividades; (iii) usuários que retomaram atividades pendentes; (iv) usuários que concluíram a atividade, e, por fim, (v) monitoramento de uso dos *kits*

¹² <http://cpntools.org>

¹³ O Instituto Universidade Virtual (UFC Virtual) que co-coordena os cursos semi-presenciais da Universidade Aberta do Brasil (UAB), junto a Universidade Federal do Ceará (UFC), trabalha com um professor titular da disciplina (responsável por todos os pólos), um tutor a distância e um tutor presencial. Vide página <http://www.virtual.ufc.br>.

¹⁴ Cada movimentação de fichas em um modelo é contabilizado em um passo. Assim, têm-se vários passos por simulação.

(identificação dos equipamentos e componentes utilizados pelos alunos) e bancadas (localização do aluno).

A validação e verificação dos requisitos necessitam de ações que estejam bem claras e definidas na Rede de Petri Colorida. O fluxo dos usuários, modelado através das dinâmicas das fichas em RdPC, devem ser os compatíveis com os especificados no Diagrama de Atividades do Fluxo de Realização de Atividades por professor e por aluno (Figura 2). Para tanto, as estatísticas retiradas dos monitores, como médias, permitem ver se cada cenário se comportou de forma satisfatória, mesmo com um número grande de ciclos de uso do laboratório, que abrange as possibilidades de o aluno, não realizar a atividade, ter a atividade interrompida, retomar uma atividade interrompida e realizar a atividade completa. Embora haja estas possibilidades, todos os usuários deverão realizar uma vez a atividade por completo para fins de simulação. No modelo, as interrupções antes de começar as práticas e as saídas logo após a entrada no laboratório têm a mesma probabilidade dos usuários que finalizaram a atividade (50%). A análise deste comportamento permite verificar e validar, ao final, se realmente os requisitos estão funcionando. A análise de cada cenário e o comportamento do fluxo das fichas, disparo das transições e análise das redes particulares, permite uma visão de cada requisito. Nos gráficos 1 a 5, a escala do eixo y representa o número de fichas que foram depositadas no lugar da rede que está sendo monitorado.

O Mínimo é o menor valor do somatório de fichas que um lugar conteve nas 100 simulações e o Máximo é o maior valor do somatório de fichas que um lugar conteve nas 100 simulações. Já a Média por usuário (Avrg) é calculada através da seguinte fórmula¹⁵:

$$\sum_{i=1}^n x_i = x_1 + x_2 \dots + x_n$$

$Avrg_n = \frac{\sum_{i=1}^n x_i}{n}$, em que $Avrg_n$ será a Média usada (Avrg), x_n é número de vezes que um Monitor é ativado para determinada ficha. Já n é o número de simulações.

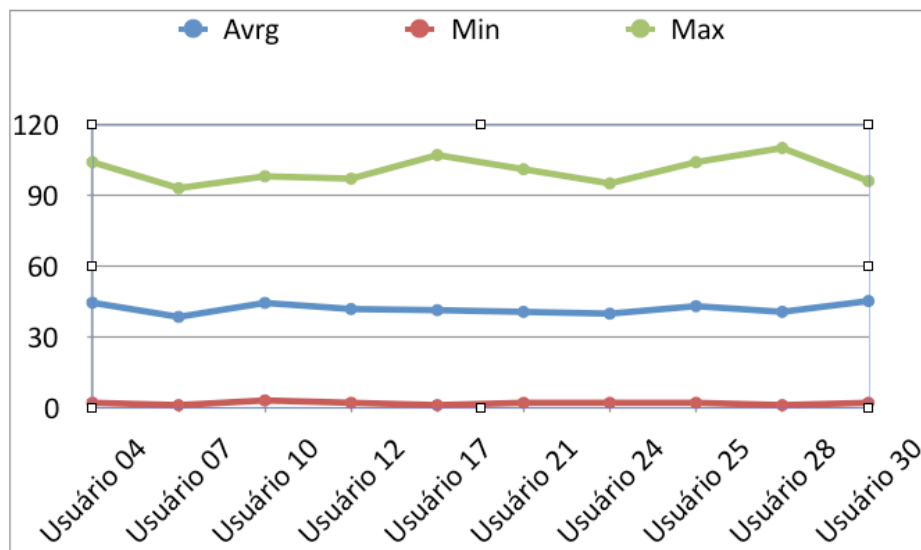
¹⁵ http://cpntools.org/documentation/tasks/performance/calculating_statistics

4.1 Estatísticas de entradas dos alunos no u-LabPA

Os dados analisados nesta parte dizem respeito à entrada dos alunos no laboratório, valor representado pelo eixo Y apresentados no Gráfico 1. O Monitor foi inserido na transição “Solicita token” que está no modelo RdPC apresentado na Figura 13.

As informações vistas no gráfico mostram que o comportamento das Médias dos usuários monitorados é próximo ao linear, não havendo, portanto, perturbações que pudessem levar a possíveis problemas no sistema, tais como *loop* infinitos. Assim, o comportamento do requisito de “Autenticação”, modelado nesta etapa se mostrou consistente.

Gráfico 1 - Estatísticas de entradas dos alunos no u-LabPA



Fonte: elaborado pelo autor.

4.2 Estatísticas das saídas dos alunos logo após a entrada no u-LabPA

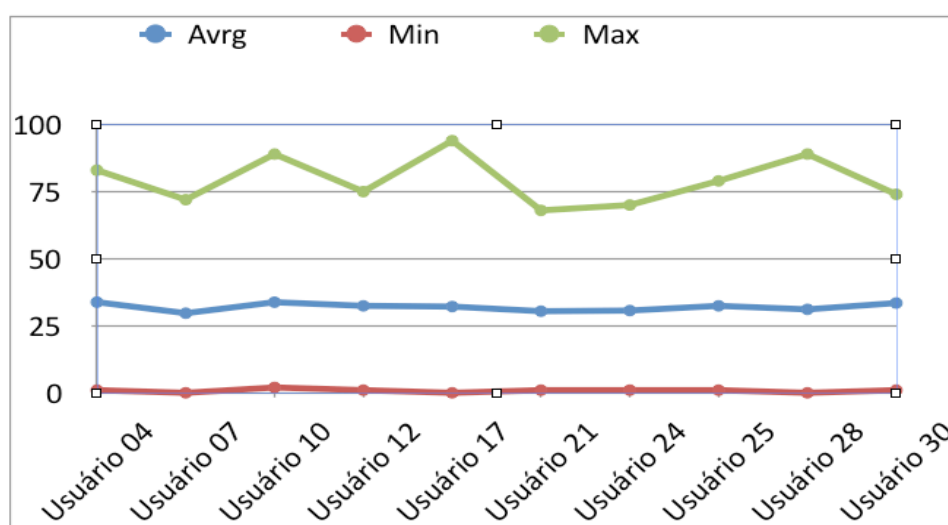
Esta etapa de execução das redes também trata do requisito “Autenticação”, mas no que diz respeito a saída do usuário do sistema. No Gráfico 2 são mostrados os dados provenientes dos dez usuários que foram monitorados no processo de “Saída dos alunos logo após a entrada no u-LabPA”, que é definido como a identificação de saídas do aluno no sistema antes de iniciar sua atividade. No eixo Y do gráfico são contabilizados os valores de

saídas do laboratório. Os dados colhidos são, a primeira vista, elevados se comparados a um ambiente real porém isto se deve a dois fatores. Primeiro, foi deixado o fluxo de usuários livres – sem regras restritivas, como somente um número x de entradas possíveis no laboratório -, e observado somente seu comportamento geral, no modelo. Segundo, a probabilidade de saída antes da finalização da atividade e a finalização da atividade são as mesmas, ou seja 50%, para cada. O Monitor foi inserido na transição “Usuario resolve sair” do modelo RdPC visto na Figura 11.

No Gráfico 1 é observada a entrada dos usuários no u-LabPA, onde se percebe uma média de entrada em torno de 42 por aluno, já no Gráfico 2 pode-se observar a saída destes usuários logo após a entrada no laboratório, percebe-se que dos 42 *login* efetuados, 32 saíram logo em seguida, esse resultado indica que se registram mais 9 saídas após o usuário ter uma atividade atribuída a ele, já que esta modelagem está limitada a conclusão de 1 atividade por aluno. Pode-se observar este comportamento no Gráfico 3 (Estatísticas das Saídas dos Alunos Sem Concluir a Atividade)¹⁶.

O comportamento do sistema, que pode ser visto em relação a média das Médias, está sem anomalias. O requisito de “Autenticação” se comportou de forma satisfatória.

Gráfico 2 - Estatísticas das saídas dos alunos logo após a entrada no u-LabPA



Fonte: elaborado pelo autor.

¹⁶ No caso, é considerado a media das medias dos valores.

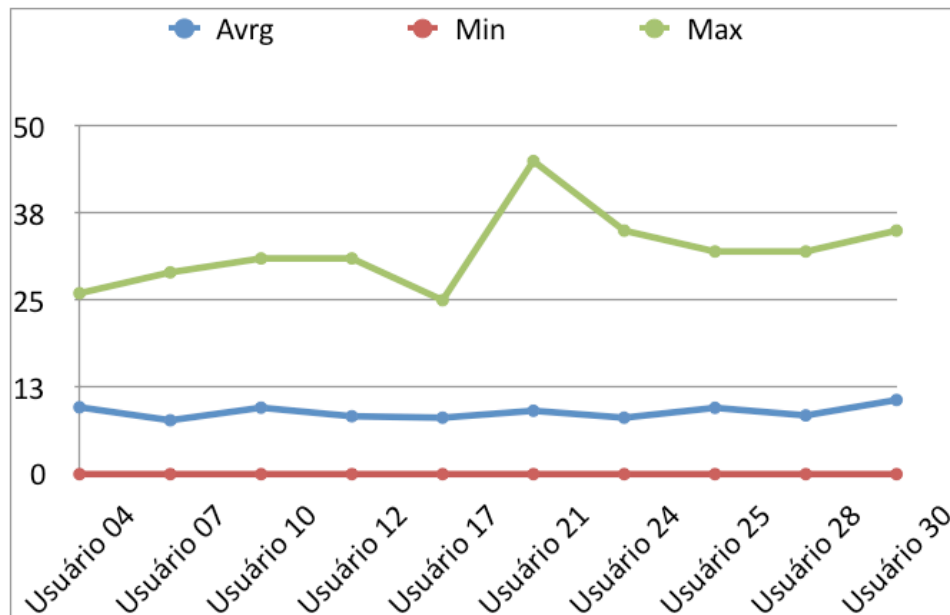
4.3 Estatísticas das saídas dos alunos sem concluir a atividade

Nesta etapa de execução das redes, os requisitos de “Autenticação”, “Cadastra Atividade”, “Registra Atividade do Usuário” e “Realiza Atividade” são testados. No Gráfico 3 são compilados pelo Monitor os dados referentes aos alunos que saíram do laboratório com uma atividade atribuída, porém, sem ter sua prática laboratorial concluída. O Monitor utilizado para coletar os dados deste gráfico foi inserido na transição “Usuario decide deixar a sala” que está no modelo de RdPC que pode ser visto da Figura 11.

O Gráfico 3 possui o valor médio das Médias igual a 9, valor que como visto no final do item 4.2, também está relacionado aos *logout* do sistema. Estes são os *logout* – saídas – sem conclusão das atividades. Com estes, portanto, ficam 41 *logout* do sistema. Ficam, assim, registradas todas as saídas sem conclusão de atividades nos gráficos monitorados. Percebe-se que não houve conclusão de atividades até este ponto monitorado.

Os requisitos se mostraram coerentes, sendo seguido cada passo neles definidos.

Gráfico 3 - Estatísticas das saídas dos alunos sem concluir a atividade



Fonte: elaborado pelo autor.

4.4 Estatísticas das atividades retomadas pelos alunos

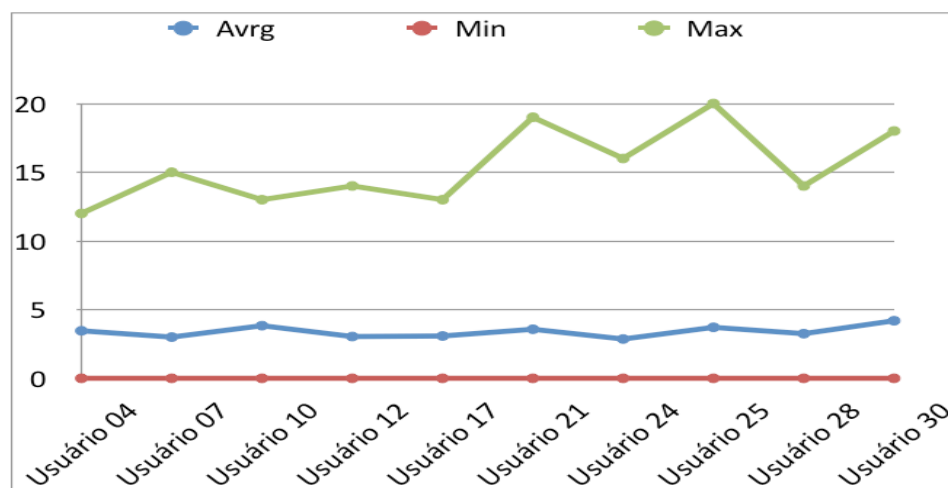
Outro aspecto monitorado foi o número de atividades retomadas (Gráfico 4), cujo Monitor foi colocado na transição “Aluno retomando atividade” do modelo da RdPC que pode ser vista na Figura 14. Neste caso, os requisitos de “Autenticação”, “Cadastra Atividade”, “Registra Atividade do Usuário” e “Realiza Atividade” são verificados novamente. O aluno pode retomar a atividade do ponto onde estava ou iniciar uma nova.

No Gráfico 4 é mostrado o número de vezes que um aluno retoma sua atividade, percebe-se que em média, 3 atividades são retomadas por aluno. No Gráfico 3 é mostrado que se teve no sistema, em média, 9 saídas com atividade alocada por aluno. Nesta etapa do modelo, o usuário poderá ter uma atividade nova atribuída ou retomar uma atividade que deixou pendente anteriormente. Após receber a sua atividade, o aluno poderá seguir para concluí-la ou sair novamente. Como pode ser visto no Gráfico 3, foram constatadas em média, cada aluno saiu nove vezes sem concluir a atividade. Estas poderiam ser atividades iniciadas que estariam também sendo contabilizadas nas atividades retomadas mostradas no Gráfico 4.

Por fim, há uma relação entre o valor médio das Médias de usuários que são vistas nos Gráficos 3 e 4. Em média, das nove saídas dos usuários que foram computadas, duas a três são relacionadas às saídas com atividades retomadas.

Os requisitos modelados nas redes simuladas nesta etapa se mostraram válidos.

Gráfico 4 - Estatísticas das atividades retomadas pelos alunos



Fonte: elaborado pelo autor.

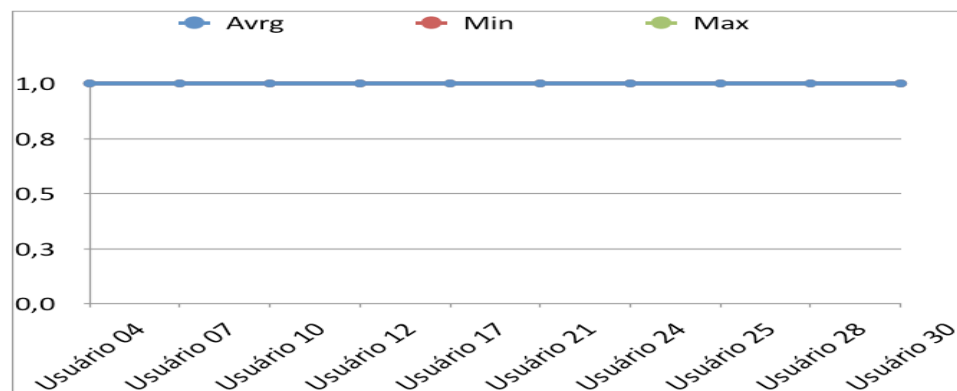
4.5 Estatísticas das alocações e das atividades finalizadas pelos alunos

Analisando-se as alocações de bancadas, instrumentos e componentes (Gráfico 5) e de atividades finalizadas (Gráfico 6), percebe-se que os valores de Máximo, Mínimo e Média são iguais a 1, ou seja, todos os indivíduos monitorados realizaram a atividade por completo, conforme regra estabelecida na simulação.

Como a finalização da atividade engloba a alocação de bancadas, instrumentos e componentes, o sistema apresentou exatamente o comportamento esperado.

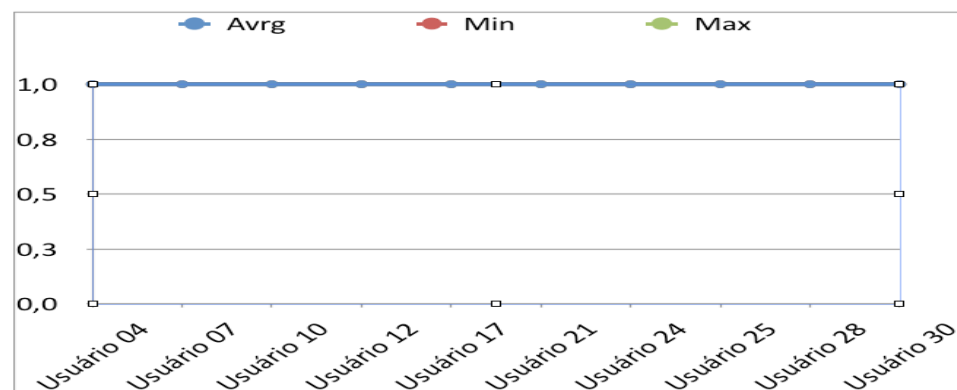
Os Monitores foram colocados nas transições “Usuario pega instrumentos, bancada e componentes”, no caso do Gráfico 5, e “finalizar”, no caso do Gráfico 6. Estas transições estão, respectivamente, nos modelos das RdPC vistas nas Figuras 15 e 16.

Gráfico 5 - Estatísticas das alocações



Fonte: elaborado pelo autor.

Gráfico 6 - Estatísticas das atividades finalizadas pelos alunos



Fonte: elaborado pelo autor.

O comportamento das redes e sua relação com os requisitos, após o processo de validação e refinamento destes, permitiu que se pudesse ter uma verificação completa do comportamento dos requisitos de “Autenticação”, “Cadastra Atividade”, “Registra Atividade de Usuário”, “Disponibiliza Atividades de Usuário” e “Realiza Atividade”. Isto permitiu reforçar o processo de Engenharia de Requisitos do sistema e desenvolvimento do *framework*.

5 CONCLUSÕES

A temática deste trabalho é a modelagem formal em Redes de Petri Coloridas dos requisitos relacionados a Realização de Atividades do *framework* u-LabPA proposto por Sarmiento (2016). A análise formal contemplou o grupo de requisitos elencados para a prática de laboratório, “Autenticação”, “Cadastra Atividade”, “Registra Atividade de Usuário”, “Disponibiliza Atividades de Usuário” e “Realiza Atividade”, presentes no *framework* u-LabPA; bem como o fluxo de atividades que compreende o processo chamado de “Realização de Atividade”, feita pelo aluno, que envolve não só o *framework* mas aplicações feitas a partir deste (e.g. Ambiente de Atividade do Aluno e ambiente de Realidade Virtual LaVES). Em ambos os casos uma linguagem formal é importante para ser utilizada a fim de ver tanto a corretude (verificação), quanto a completude (validação) dos requisitos e comportamento do modelo do *software*, conforme foi visto no capítulo de referencial teórico.

As Redes de Petri Coloridas foram utilizadas para implementar os Casos de Uso em forma de cenários de simulação, permitindo visualizar a dinâmica do funcionamento do sistema, mesmo antes de seu desenvolvimento. Tal característica permitiu que fosse possível extrair informações complementares que não constavam nos casos de uso iniciais do cliente. As informações complementares foram:

- Maior detalhamento do Fluxo de Realização de Atividades realizado pelo aluno (e.g. modificação dos requisitos “Identifica contexto de aluno” e “Disponibiliza dados de usuário”);
- Criação dos Casos de Uso “Disponibiliza atividades de usuário” e “Verifica contexto de laboratório” (relacionados indiretamente com os Casos de Uso modelados);
- Remoção de parte do detalhamento do Caso de Uso “Realiza Atividade” que foi especificado em outros casos.

Estas modificações foram importantes para que a modelagem pudesse ser representada o conjunto de Casos de Uso mais próximos ao sistema desejado pelo cliente e, portanto, refletindo suas reais necessidades.

Como *framework*, o u-LabPA não poderia ter seus requisitos testados sem que fossem feitos modelos de aplicações que usassem suas características, como foi o caso da criação das aplicações AAA, AAP e RV (que usa o LaVES). Estes *softwares* usariam as funcionalidades do *framework* para que alunos e professores pudessem acessar as atividades. Esta é outra peça importante percebida durante o processo de modelagem, onde na camada do

u-LabPA fica mais clara de se perceber em relação às aplicações que a utilizam e onde há a relação com o *hardware*, conforme pode ser visto na Figura 08.

É importante esclarecer que a parte do *framework* que trata das informações para a simulação posterior à prática pode ser acessado por aplicações externas, como no caso de uma aplicação de Realidade Virtual. Porém o *framework* não oferece recursos para a construção deste tipo de *software*.

As simulações feitas para validação e verificação do sistema permitiram ver o comportamento de cada requisito em separado e em conjunto, conforme foi analisado no Capítulo 4. O comportamento das RdPC, precisou ser ajustado a fim de permitir que o ciclo de realização das atividades dos usuários nas simulações fosse concluído, possibilitando a identificação dos seguintes problemas:

- Manutenção do contexto dos alunos;
- Persistência de informação dos alunos;
- Falta de uma melhor especificação sobre os limites de uso do laboratório por parte do aluno e sobre o tempo de conclusão para uma atividade. Neste caso, o sistema carecia de maior refinamento para identificar o problema de vagas em Laboratório X Tempo de Execução da Atividade;
- Teste para uma função de criptografia de modo genérico (criptografia simples por substituição). Percebeu-se que seria melhor implementar uma forma de suportar diversas abordagens mais seguras (e.g. *Advanced Encryption Standard* ou *Triple Data Encryption Standard*, como seria feito no *framework*).

Os problemas verificados na modelagem formal em RdPC permitiram identificar situações que só seriam identificadas no decorrer da codificação, com o fluxo dos dados. A identificação tanto visual quanto através dos dados advindos dos monitores foi fundamental para a identificação e correção destes problemas. A RdPC permitiu verificar que o sistema carecia de uma melhor política de controle de acesso ao laboratório e de alocação de tempo do usuário (aluno) para execução de sua prática.

Por fim, do processo de elicitação dos requisitos usando RdPC, é possível ir para a codificação de forma mais satisfatória, minimizando o número de erros do código gerado.

5.1 Trabalhos futuros

A ferramenta utilizada no processo de modelagem das RdPC - *CPN Tools* - gera os modelos criados no formato *Extensible Markup Language* (XML), o que possibilita que

aplicações externas automatizem a extração de classes a partir destes modelos e sejam criados protótipos codificados dos requisitos testados. Estes protótipos podem ser aferidos em ambiente real e ter seus resultados comparados com os resultados obtidos pelas simulações feitas. Tal processo permitirá saber o quão próximo do real as redes chegaram em seu comportamento. Esta seria a primeira fase para continuação do processo de verificação e validação formal de todo o *framework* u-LabPA a fim de se ter todos os cenários dos requisitos deste *framework*, com suas respectivas simulações, e desta forma poder melhorá-lo e verificar possíveis falhas. Como o sistema se encontra em processo concomitante de desenvolvimento, a abordagem feita neste trabalho poderá convergir para o trabalho de Sarmiento (2016) e auxiliar na parte de validação e verificação de seus requisitos.

REFERÊNCIAS

ARANTES, Valeria (Org.). **Educação a Distância: Pontos e Contrapontos**. São Paulo: Summus, 2011.

ARAÚJO, Regina Borges. Computação Ubíqua: Princípios, Tecnologias e Desafios. **Xxi Simpósio Brasileiro de Redes de Computadores (sbrc 2003)**, São Carlos, Sp, p.44-115, 2003. Disponível em: <http://professordiovani.com.br/rw/monografia_araujo.pdf>. Acesso em: 15 nov. 2015.

ARAÚJO, Régia T. S et al. Modelagem de um sistema de gestão na Educação a Distância no Brasil utilizando redes de Petri Coloridas. **Ingeniare. Revista Chilena de Ingeniería**, [s.l.], v. 23, n. 1, p.145-158, 1 jan. 2015. SciELO Comision Nacional de Investigacion Cientifica Y Tecnologica (CONICYT). Disponível em: <<http://dx.doi.org/10.4067/s0718-33052015000100016>>. Acesso em: 22 Out. 2015.

BARBOSA, Débora Nice Ferrari et al. Em direção a Educação Ubíqua: aprender sempre, em qualquer lugar, com qualquer dispositivo. **Revista Renote: Novas tecnologias na educação**, Rio Grande do Sul, v. 6, n. 1, p.1-11, dez. 2008. Disponível em: <<http://seer.ufrgs.br/index.php/renote/article/view/14492>>. Acesso em: 10 nov. 2015.

BARGAOUI, Hichem; BDIWI, Rawia. Smart classroom: Design of a gateway for ubiquitous classroom. **2014 International Conference On Web And Open Access To Learning (icwoal)**, [s.l.], p.1-4, nov. 2014. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/icwoal.2014.7009206>. Disponível em: <<http://ieeexplore.ieee.org/document/7009206/>>. Acesso em: 5 out. 2016.

CARDOSO, J.; VALETTE, R. **Redes de Petri**. Florianópolis: UFSC, 1997.

CHALMERS, Dan. Pervasive Computing as a Classroom-Based Course. **Ieee Pervasive Computing**, [s.l.], v. 14, n. 3, p.70-78, jul. 2015. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/mprv.2015.45>. Disponível em: <<http://ieeexplore.ieee.org/document/7140699/>>. Acesso em: 29 set. 2016.

CHIN, Jeannette; CALLAGHAN, Vic. Educational Living Labs: A Novel Internet-of-Things Based Approach to Teaching and Research. **2013 9th International Conference On Intelligent Environments**, [s.l.], p.92-99, jul. 2013. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/ie.2013.48>. Disponível em: <<http://ieeexplore.ieee.org/document/6597796/>>. Acesso em: 02 out. 2016.

CHOOSANG, San; GORDON, Steven. A Coloured Petri Net Methodology and Library for Security Analysis of Network Protocols. **Journal Of Computers**, [s.l.], v. 9, n. 2, p.243-256, 1 fev. 2014. International Academy Publishing (IAP). <http://dx.doi.org/10.4304/jcp.9.2.243-256>. Disponível em: <<https://sandilands.info/sgordon/doc/choosang2014-coloured.pdf>>. Acesso em: 10 out. 2016.

CIARDELLI, Lorenzo; BIXIO, Luca; REGAZZONI, Carlo S.. Interaction modeling in automotive applications: A cognitive approach. **2011 Ieee International Multi-disciplinary Conference On Cognitive Methods In Situation Awareness And Decision Support (cogsima)**, [s.l.], p.248-251, fev. 2011. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/cogsima.2011.5753453>. Disponível em: <<http://ieeexplore.ieee.org/document/5753453/>>. Acesso em: 25 set. 2016.

CIRILO, Carlos Eduardo. **Computação ubíqua: definição, princípios e tecnologias**. 2007, p. 6. Disponível em:<http://ufscar.academia.edu/ducirilo/Teaching/14124/Computacao_Ubiqua_definicao_principios_e_tecnologias>. Acesso em: 06 set. 2016.

DEY, Anind K.; ABOARD, Gregory D.. CybreMinder: A Context-Aware System for Supporting Reminders. **Proceedings of the second international symposium on Handheld and Ubiquitous Computing (HUC 2K)**. Bristol, Uk, p.172-186, set. 2000. Disponível em: <<http://dl.acm.org/citation.cfm?id=757284>>. Acesso em: 05 nov. 2016.

FERNANDES, Carla Ferreira. **Verificação e Refinamento de Requisitos em Árvore de Características usando Linhas de Produtos de Requisitos e Redes de Petri**. 2012. 74 f. TCC (Graduação) - Curso de Bacharelado em Ciência da Computação, Universidade Federal de Goiás – Ufg, Catalão, 2012. Disponível em: <<https://dcc.catalao.ufg.br/up/498/o/CarlaFerreira2012.pdf>>. Acesso em: 13 set. 2016.

FISCHER, Maria Clara Barros de Oliveira; SILVA, Flávio Soares Corrêa da. **Estudo de requisitos para um software educativo de apoio ao ensino da introdução a computação**. 2001. 170 f. Dissertação (Mestrado) - Ciência da Computação, Instituto de Matemática e Estatística da Universidade de São Paulo, São Paulo, 2001. Disponível em: <https://www.researchgate.net/publication/35489432_Estudo_de_requisitos_para_um_software_educativo_de_apoio_ao_ensino_da_introducao_a_computacao>. Acesso em: 29 set. 2016.

GEHLOT, Vijay; NIGRO, Carmen. An introduction to systems modeling and simulation with Colored Petri Nets. **Proceedings Of The 2010 Winter Simulation Conference**, [s.l.], p.104-118, dez. 2010. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/wsc.2010.5679170>. Disponível em: <<http://ieeexplore.ieee.org/document/5679170/>>. Acesso em: 18 set. 2015.

HAMDAOUI, Bechir; ALSHAMMARI, Tamara; GUIZANI, Mohsen. Exploiting 4G mobile user cooperation for energy conservation: challenges and opportunities. **Ieee Wireless Communications**, [s.l.], v. 20, n. 5, p.62-67, out. 2013. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/mwc.2013.6664475>. Disponível em: <<http://ieeexplore.ieee.org/document/6664475/>>. Acesso em: 27 nov. 2015.

JENSEN, Kurt. **Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use**. Vol 1. New York: Springer Publishing Company, 2010.

KIM, Dongjin et al. Improving cluster tools performance using Colored Petri Nets in semiconductor manufacturing. **Proceedings Title: Proceedings of the 2012 Winter Simulation Conference (WSC)**, [s.l.], dez. 2012. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/wsc.2012.6465316>. Disponível em: <<http://ieeexplore.ieee.org/document/6465316/>>. Acesso em: 3 nov. 2015.

MA, Longhua et al. Net-in-Net: Interaction Modeling for Smart Community Cyber-Physical System. **2010 7th International Conference On Ubiquitous Intelligence & Computing And 7th International Conference On Autonomic & Trusted Computing**, [s.l.], p.250-255, out. 2010. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/uic-atc.2010.15>. Disponível em: <<http://ieeexplore.ieee.org/document/5667190/>>. Acesso em: 13 nov. 2016.

MARTINIE, Célia; NAVARRE, David; PALANQUE, Philippe. A multi-formalism approach for model-based dynamic distribution of user interfaces of critical interactive systems. **International Journal Of Human-computer Studies**, [s.l.], v. 72, n. 1, p.77-99, jan. 2014. Elsevier BV. <http://dx.doi.org/10.1016/j.ijhcs.2013.08.013>. Disponível em: <<http://dl.acm.org/citation.cfm?id=2562543>>. Acesso em: 22 set. 2016.

MIRLACHER, Thomas; PALANQUE, Philippe; BERNHAUPT, Regina. Engineering animations in user interfaces. **Proceedings Of The 4th Acm Sigchi Symposium On Engineering Interactive Computing Systems - Eics '12**, [s.l.], p.111-120, 2012. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/2305484.2305504>. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2305484.2305504>>. Acesso em: 22 set. 2016.

MORAN, J. M.; MASETTO, M.; BEHRENS, M. **Novas Tecnologias e Mediação pedagógica**. 21. ed. São Paulo: Papirus, 2013.

NGUYEN, Hai Quang; APON, Amy. Parallel file system measurement and modeling using colored petri nets. **Proceedings Of The Third Joint Wosp/sipew International Conference On Performance Engineering - Icppe '12**, [s.l.], p.30-30, 2012. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/2188286.2188321>. Disponível em: <<http://dl.acm.org/citation.cfm?doid=2188286.2188321>>. Acesso em: 29 out. 2016.

OLIVEIRA, Cintia Carvalho. **Uma análise de metodologias formais, baseadas em Redes de Petri para modelagem de software**. 2010. 12 f. Dissertação (Mestrado) - Curso de Programa de Pós-graduação em Ciência da Computação, Ciência da Computação, Universidade Federal de Uberlândia (ufu), Uberlândia, Mg, 2011. Disponível em: <<http://www.facom.ufu.br/posgrad/wd2010/artigos/CintiaOliveira.pdf>>. Acesso em: 30 set. 2016.

OLIVEIRA, Ricardo B. D.; COSTA, Diego P.. Benefícios da Computação Pervasiva na Educação e mobUS, Um Sistema Móvel no Auxílio à Aprendizagem. **Revista de Informática Aplicada**, São Caetano do Sul, v. 10, n. 2, 2014. Disponível em: <<http://www.ria.net.br/index.php/ria/article/viewFile/128/142>>. Acesso em: 5 out. 2016.

PAILLARD, G. A. L. et al. Extended MOBILIS. **Proceedings Of The 6th Euro American Conference On Telematics And Information Systems - Eatis '12**, [s.l.], 2012. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/2261605.2261668>. Disponível em: <<http://ieeexplore.ieee.org/document/6218012/>>. Acesso em: 18 set. 2016.

PEIXOTO, Maria Joelma Pereira; CARVALHO, Windson Viana de; SARMENTO, Wellington Wagner Ferreira. Uma Proposta de Ferramenta de Apoio ao Ensino Prático de Computação Ubíqua. **Webmedia2016: XXII Simpósio Brasileiro de Sistemas Multimídia e Web**, Teresina, Pi, p.17-22, nov. 2016. Disponível em:

<http://www6.ifpi.edu.br/webmedia/wp-content/uploads/2016/11/WebMedia2016__Anais__Volume2__Workshops_e_Posteres.pdf>. Acesso em: 12 nov. 2016.

PEQUENO FILHO, Paulo de Tarso Cavalcante et al. U-LabPA: A framework based on ubiquitous computing for laboratory activities. **2016 8th Euro American Conference On Telematics And Information Systems (eatis)**, [s.l.], abr. 2016. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/eatis.2016.7520146>. Disponível em: <<http://ieeexplore.ieee.org/document/7520146/>>. Acesso em: 10 nov. 2016.

PETERSON, J. L. **Petri Net Theory and the Modeling of Systems**. New York: McGraw-Hill, 1981.

RICHARDS, Mike; PETRE, Marian; BANDARA, Arosha K.. Starting with Ubicomp. **Proceedings Of The 43rd Acm Technical Symposium On Computer Science Education - Sigese '12**, [s.l.], p.583-588, 2012. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/2157136.2157306>. Disponível em: <<http://dl.acm.org/citation.cfm?id=2157306>>. Acesso em: 14 set. 2016.

SARMENTO, Wellington Wagner Ferreira. **Integração de um Ambiente Virtual de Aprendizagem com Aplicações Móveis de Suporte a Educação a Distância**. 2007. 141 f. Dissertação (Mestrado) - Curso de Graduação em Engenharia de Teleinformática, Departamento de Engenharia de Teleinformática, Universidade Federal do Ceará - Ufc, Fortaleza - Ce, 2007. Disponível em: <<http://www.repositorio.ufc.br/handle/riufc/16111>>. Acesso em: 23 set. 2016.

SARMENTO, W. W. F. et al. U-Lab. **Proceedings Of The 6th Euro American Conference On Telematics And Information Systems - Eatis '12**, [s.l.], p.260-266, 2012. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/2261605.2261644>. Disponível em: <<http://ieeexplore.ieee.org/document/6218031/>>. Acesso em: 17 out. 2016.

SARMENTO, Wellington Wagner Ferreira. **Um framework baseado em Computação Ubíqua para Atividades Laboratoriais com suporte a Redes de Sensores Sem-Fio e Identificação por Rádio Frequência integrado a Ambientes Virtuais de Aprendizagem**. 2016. Draft da Tese (Doutorado) - Curso de Doutorado em Engenharia de Teleinformática, Pós-graduação em Engenharia de Teleinformática, Universidade Federal do Ceará, Fortaleza - Ce, 2016.

SILVA, Juarez Bento da et al. A DC Electric Panel Remote Lab. **International Journal Of Online Engineering (ijoe)**, [s.l.], v. 12, n. 04, p.30, 28 abr. 2016. International Association of Online Engineering (IAOE). <http://dx.doi.org/10.3991/ijoe.v12i04.5096>. Disponível em: <https://www.researchgate.net/publication/301717607_A_DC_electric_panel_remote_lab>. Acesso em: 20 out. 2016.

SOMMERVILLE, I. **Engenharia de Software**. 6. Ed. São Paulo: Pearson, 2005.

SOMMERVILLE, I. **Engenharia de Software**. 9. Ed. São Paulo: Pearson, 2011.

SOUZA, Maria de Fátima Costa de et al. LOCPN: Redes de Petri Coloridas na Produção de Objetos de Aprendizagem. **Revista Brasileira de Informática na Educação (rbie)**, Brasil, v. 15, n. 3, p.39-42, 2007. Disponível em: <<http://www.proativa.vdl.ufc.br/publicacoes/artigos/e048ba06c5fb169f7af1a85865a75522.pdf>>. Acesso em: 13 out. 2016.

TANENBAUM, Andrew S. **Modern Operating Systems**. 3. Ed. Prentice-Hall: Pearson, 2008.

WEIDLICH, Matthias; MENDLING, Jan; GAL, Avigdor. Net-Based Analysis of Event Processing Networks – The Fast Flower Delivery Case. **Application And Theory Of Petri Nets And Concurrency**, [s.l.], p.270-290, 2013. Springer Nature. http://dx.doi.org/10.1007/978-3-642-38697-8_15. Disponível em: <<http://dl.acm.org/citation.cfm?id=2524387>>. Acesso em: 15 set. 2016.

WEISER, Mark. Some computer science issues in ubiquitous computing. **Communications Of The Acm**, [s.l.], v. 36, n. 7, p.75-84, 1 jul. 1993. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/159544.159617>. Disponível em: <<http://dl.acm.org/citation.cfm?id=159617>>. Acesso em: 23 out. 2016.

WIKIPEDIA. **Rede de Petri**. Verbete da Wikipedia sobre Redes de Petri, 2016. Disponível em: <https://pt.wikipedia.org/wiki/Rede_de_Petri>. Acesso em: 11 Nov. 2016.

WOODCOCK, Jim et al. Formal methods. **Acm Computing Surveys**, [s.l.], v. 41, n. 4, p.1-36, 1 out. 2009. Association for Computing Machinery (ACM). <http://dx.doi.org/10.1145/1592434.1592436>. Disponível em: <<http://dl.acm.org/citation.cfm?doid=1592434.1592436>>. Acesso em: 25 set. 2016.

APÊNDICE A – TABELA DE DADOS PROVENIENTES DOS MONITORES

CPN Tools Performance Report			
Net: //vmware-host/Shared Folders/Mesa/Artigo cp			
Number of replications: 100			
Entrou no u-LabPA			
Name	Avrg	Min	Max
Usuário 04	44	2	104
Usuário 07	38	1	93
Usuário 10	44	3	98
Usuário 12	42	2	97
Usuário 17	41	1	107
Usuário 21	41	2	101
Usuário 24	40	2	95
Usuário 25	43	2	104
Usuário 28	41	1	110
Usuário 30	45	2	96
Saiu do u-LabPA			
Name	Avrg	Min	Max
Usuário 04	34	1	83
Usuário 07	30	0	72
Usuário 10	34	2	89
Usuário 12	32	1	75
Usuário 17	32	0	94
Usuário 21	30	1	68
Usuário 24	31	1	70
Usuário 25	32	1	79
Usuário 28	31	0	89
Usuário 30	33	1	74
Saiu sem concluir a atividade			
Name	Avrg	Min	Max
Usuário 04	10	0	26
Usuário 07	8	0	29
Usuário 10	10	0	31
Usuário 12	8	0	31
Usuário 17	8	0	25
Usuário 21	9	0	45
Usuário 24	8	0	35
Usuário 25	10	0	32
Usuário 28	8	0	32
Usuário 30	11	0	35

Generated: Mon Nov 28 13:45:40 2016			
Retoma atividade			
Name	Avrg	Min	Max
Usuário 04	3	0	12
Usuário 07	3	0	15
Usuário 10	4	0	13
Usuário 12	3	0	14
Usuário 17	3	0	13
Usuário 21	4	0	19
Usuário 24	3	0	16
Usuário 25	4	0	20
Usuário 28	3	0	14
Usuário 30	4	0	18
Pega o kit e aloca uma bancada			
Name	Avrg	Min	Max
Usuário 04	1,0	1,0	1,0
Usuário 07	1,0	1,0	1,0
Usuário 10	1,0	1,0	1,0
Usuário 12	1,0	1,0	1,0
Usuário 17	1,0	1,0	1,0
Usuário 21	1,0	1,0	1,0
Usuário 24	1,0	1,0	1,0
Usuário 25	1,0	1,0	1,0
Usuário 28	1,0	1,0	1,0
Usuário 30	1,0	1,0	1,0
Finalizou a Atividade			
Name	Avrg	Min	Max
Usuário 04	1,0	1,0	1,0
Usuário 07	1,0	1,0	1,0
Usuário 10	1,0	1,0	1,0
Usuário 12	1,0	1,0	1,0
Usuário 17	1,0	1,0	1,0
Usuário 21	1,0	1,0	1,0
Usuário 24	1,0	1,0	1,0
Usuário 25	1,0	1,0	1,0
Usuário 28	1,0	1,0	1,0
Usuário 30	1,0	1,0	1,0

Fonte: elaborada pelo autor.


```

▼ Project declarations
  ▶ val XMLObjetos
  ▶ val BIBObjetos
  ▶ val BIBconexoes
  ▶ colset ID = INT;
  ▶ colset NOME = string;
  ▶ colset SEMESTRE = real;
  ▶ colset DESC = string;
  ▶ colset XML_OBJ
  ▶ colset XML_OBJ_IN
  ▶ colset XML_OBJ_OUT = record Objetos : XML_OBJ_IN * Aluno : crypto;
  ▶ colset XML_OBJIS = list XML_OBJ;
  ▶ colset XML_INFO = record id_atividade : ID * nome_atividade : NOME * desc_atividade : DESC * Id_aluno : INT * semestre : SEMESTRE;
  ▶ colset XML = record Info : XML_INFO * Objetos : XML_OBJ_IN * Aluno : crypto;
  ▶ colset OBJ_ID = ID;
  ▶ colset OBJ_NOME = string;
  ▶ colset CIV_TIPO = ID;
  ▶ colset CIV_NOME = STRING;
  ▶ colset CIV_DEST = ID;
  ▶ colset CEN_OBJIS
  ▶ colset BIB_OBJIS = product OBJ_ID * OBJ_NOME;
  ▶ colset BIB_CIV = product CIV_TIPO * CIV_NOME;
  ▶ colset RELAT = record Info : XML_INFO * Objetos : CEN_OBJIS;
  ▶ var xml:XML;
  ▶ var xml_info:XML_INFO;
  ▶ var xml_obj:XML_OBJ;
  ▶ var xml_obj_in:XML_OBJ_IN;
  ▶ var xml_obj_out:XML_OBJ_OUT;
  ▶ var xml_objis:XML_OBJIS;
  ▶ var o1,d1,c1: OBJ_ID;
  ▶ var o2,d2,c2: OBJ_NOME;
  ▶ var objeto: CEN_OBJIS;
  ▶ var sim:REAL;
  ▶ var result:STRING;
  ▶ var relat:RELAT;
  ▶ fun Avaliar(x,y) =
    if (y/>1.2)
      then 1 ("Valor acima do esperado")
    else if (y/<0.9)
      then 1 ("Valor abaixo do esperado")
    else
      1 ("Valor dentro dos parametros definidos");
  ▶ fun Simular() =
    uniform(8,0,25,0);
▼ Monitors
  ▶ usuario 04 entrou no ULabPA

```

Fonte: elaborada pelo autor.

ANEXO A – FUNCIONALIDADES

F001 Autenticação

Descrição	São recebidos valores para autenticação do usuário, login e senha ou crachá e senha, e são devolvidas as permissões, caso existam, ou null, caso não haja autenticação.
Prioridade	1
Fluxo principal	<p>(a) <i>Framework</i> recebe dados para autenticação de usuário;</p> <p>(b) Verifica se autenticação pode ser feita;</p> <p>(c) Em caso positivo, retorna conjunto de permissões do usuário;</p> <p>(d) Caso contrário, retorna null.</p>

F002 Cadastra usuário

Descrição	São recebidos os dados de usuário e é realizado o cadastro, retornando erro caso haja falha.
Prioridade	2
Fluxo principal	<p>(a) <i>Framework</i> recebe dados necessários para cadastrar um novo usuário;</p> <p>(b) Caso o cadastro seja feito com sucesso, retorna ID do novo usuário.</p>
Fluxo alternativo	(a) <i>Framework</i> recebe dados para alterar usuário.
Exceções	<ul style="list-style-type: none"> ▪ Já existe um usuário cadastrado com esse cpf. ▪ Já existe um usuário cadastrado com esse email.

F003 Atribui perfil a usuário

Descrição	É atribuído o perfil passado a um dado usuário.
Prioridade	2
Fluxo principal	<p>(a) <i>Framework</i> recebe perfil e usuário;</p> <p>(b) Caso usuário não possua ainda esse perfil, atribui;</p> <p>(c) Retorna <i>true</i> caso atribuição seja ok e <i>false</i>, caso contrário.</p>
Exceções	<ul style="list-style-type: none"> ▪ Usuário já possui esse perfil. ▪ Usuário não encontrado. ▪ Perfil não encontrado.

F004 Cadastra laboratório

Descrição	São recebidos os dados de laboratório e é realizado o cadastro, retornando erro caso haja falha.
Prioridade	2
Fluxo principal	(a) <i>Framework</i> recebe dados necessários para cadastrar um novo laboratório; (b) Caso o cadastro seja feito com sucesso, retorna ID do novo laboratório.
Fluxo alternativo	(a) <i>Framework</i> recebe dados para alterar laboratório.
Exceções	<ul style="list-style-type: none"> ▪ Já existe um laboratório cadastrado com essas informações.

F005 Verifica contexto de laboratório

Descrição	Identifica <i>widgets</i> presentes em um dado laboratório.
Prioridade	1
Fluxo principal	(a) Verifica se os sensores estão operacionais (b) Verifica se os atuadores estão operacionais (c) Registra situação de sensores e atuadores do laboratório

F006 Cadastra atividade

Descrição	São recebidos os dados de atividade e é realizado o cadastro, retornando erro caso haja falha.
Prioridade	2
Fluxo principal	(c) <i>Framework</i> recebe dados necessários para cadastrar uma nova atividade; (d) Caso o cadastro seja feito com sucesso, retorna ID da nova atividade.
Fluxo alternativo	(b) <i>Framework</i> recebe dados para alterar atividade.

F007 Registra atividade de usuário

Descrição	Recebe para armazenamento dados de usuário, local, data e contexto atual da atividade sendo realizada.
Prioridade	1
Fluxo principal	(b) Recebe usuário, atividade e o que foi feito - registra individualmente cada ação do usuário relacionada a uma atividade, venha ela de uma leitura de widget ou de uma entrada manual.

F008 Disponibiliza atividades de usuário

Descrição	Retorna as atividades em que o usuário passado tem acesso.
Prioridade	2
Fluxo principal	(a) Recebe o usuário (b) Caso haja alguma atividade relacionada, retorna, assim como o status de realização de cada uma delas.
Exceções	<ul style="list-style-type: none"> ▪ Usuário não encontrado.

F009 Lê widget

Descrição	Lê status de um <i>widget</i> .
Prioridade	1
Fluxo principal	(a) Lê valor atual do <i>widget</i> solicitado.
Exceções	<ul style="list-style-type: none"> ▪ Valor não identificável.
Regras	<ul style="list-style-type: none"> ▪ Cada <i>widget</i> possui um tempo padrão de leitura que pode ser sobrescrito no cadastro do laboratório.

F010 Atua sobre widget

Descrição	Envia ação para mudança de status de um <i>widget</i> .
Prioridade	2

Fluxo principal	(a) Atualiza status do <i>widget</i> solicitado.
Exceções	<ul style="list-style-type: none"> ▪ Erro na execução do comando

F011 Disponibiliza dados de atividade de usuário

Descrição	Retorna todas as ações de um usuário em uma dada atividade.
Prioridade	2
Fluxo principal	(a) Recebe usuário e atividade. (b) Retorna todas as ações realizadas pelo usuário nessa atividade.
Exceções	<ul style="list-style-type: none"> ▪ Usuário não encontrado. ▪ Atividade não encontrada.

F012 Realiza atividade

Descrição	São realizadas as práticas propostas pelo professor para o aluno que estiver em um laboratório. Pode constar de uma ou mais práticas.
Prioridade	1
Fluxo principal	(a) Aluno abre formulário de Atividades; (b) Aluno acessa componentes, equipamento e bancadas para realização de prática (todo esse processo deverá ser monitorado pelo sistema); (c) Aluno anota os valores pedidos no formulário de Atividades; (d) Dados dos alunos e dados sensorizados de apoio são enviados para o servidor.
Exceções	<ul style="list-style-type: none"> ▪ Falha no envio de dados. Nesse caso, envia alternativamente os dados para o servidor local do laboratório para atualizar o servidor central quando for restabelecida a conexão (NF002).

F013 Identifica contexto de aluno

Descrição	Identifica status atual do aluno, hora, local, atividade.
Prioridade	1
Fluxo principal	(a) Localiza em que laboratório o aluno está fazendo sua atividade (b) Identifica em que bancada o aluno está fazendo sua atividade (c) Em que data e horário o aluno está fazendo a atividade

Fonte: Sarmiento (2016)