

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE CIÊNCIA DA COMPUTAÇÃO

Peter Prevedello

**AMBIENTE VIRTUAL COLABORATIVO PARA APOIO AO ENSINO DE
CIRCUITOS INTEGRADOS BASEADO EM EQUIPE**

Santa Maria, RS
2018

Peter Prevedello

**AMBIENTE VIRTUAL COLABORATIVO PARA APOIO AO ENSINO DE CIRCUITOS
INTEGRADOS BASEADO EM EQUIPE**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração em Ciência da Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Ciência da Computação.**

ORIENTADOR: Prof. André Luiz Aita

Santa Maria, RS
2018

Prevedello, Peter

Ambiente Virtual Colaborativo para Apoio ao Ensino de
Circuitos Integrados baseado em Equipe / Peter
Prevedello.- 2018.

99 p.; 30 cm

Orientador: André Luiz Aita

Dissertação (mestrado) - Universidade Federal de Santa
Maria, Centro de Tecnologia, Programa de Pós-Graduação em
Ciência da Computação, RS, 2018

1. Trabalho Colaborativo 2. Ambiente Virtual
Colaborativo 3. Projeto em Equipe 4. Projeto de
Circuitos Integrados em Equipe I. Aita, André Luiz II.
Título.

Sistema de geração automática de ficha catalográfica da UFSM. Dados fornecidos pelo autor(a). Sob supervisão da Direção da Divisão de Processos Técnicos da Biblioteca Central. Bibliotecária responsável Paula Schoenfeldt Patta CRB 10/1728.

©2018

Todos os direitos autorais reservados a Peter Prevedello. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

Endereço: Av Roraima, n. 1000, Prédio 7, Anexo B, Sala 287

End. Eletr.: peterprevedello@gmail.com

Peter Prevedello

**AMBIENTE VIRTUAL COLABORATIVO PARA APOIO AO ENSINO DE CIRCUITOS
INTEGRADOS BASEADO EM EQUIPE**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração em Ciência da Computação, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Mestre em Ciência da Computação.**

Aprovado em 21 de junho de 2018:

André Luiz Aita, Dr. (UFSM)
(Presidente/Orientador)

Mateus Beck Rutzig, Dr. (UFSM)

Sérgio da Costa Nunes, Dr. (IFFAR)

Santa Maria, RS
2018

DEDICATÓRIA

À minha esposa Cristiane Tirloni. Ao meu filho Pietro Tirloni Prevedello. Aos meus pais Maria Cristina dos Santos Prevedello e Percy Prevedello. Aos meus irmãos Pedro Prevedello (In Memoriam) e Percy Prevedello Junior.

AGRADECIMENTOS

Acima de tudo agradeço a Deus pela vida em plenitude.

Á minha esposa, pelo amor e pela mãe que és, cuidando do nosso amado Pietro para que eu pudesse desenvolver este trabalho e pelo incentivo de sempre nos meus estudos.

Ao meu orientador Profº Drº André Luiz Aita, pela oportunidade, confiança, incentivo e intensa contribuição no processo de elaboração deste trabalho.

RESUMO

AMBIENTE VIRTUAL COLABORATIVO PARA APOIO AO ENSINO DE CIRCUITOS INTEGRADOS BASEADO EM EQUIPE

AUTOR: Peter Prevedello
ORIENTADOR: André Luiz Aita

Este trabalho descreve a concepção e desenvolvimento de um ambiente virtual colaborativo para o apoio do ensino de projetos de circuitos integrados baseado em equipe, denominado IC-COLLAB, e sua implantação no Grupo de Microeletrônica (GMICRO) da Universidade Federal de Santa Maria (UFSM). Através de uma interface *web* simples, o ambiente virtual colaborativo fornece ferramentas de colaboração que garantem a organização dos projetos e uma cooperação eficiente entre os participantes, aperfeiçoando as práticas individuais, o trabalho em equipe e a liderança através da coordenação e comunicação. Um repositório organizado hierarquicamente permite salvar e compartilhar bibliotecas desenvolvidas individualmente pelos alunos. Um controlador de versões possibilita a todos os participantes gerenciar e documentar as mudanças realizadas nas bibliotecas. Ferramentas de comunicação permitem a troca de ideias e de conhecimento durante a realização das atividades. O compartilhamento de materiais de apoio e sugestões de projeto pode ser realizado a qualquer momento. Por fim, sempre que necessário, o ambiente permite ao coordenador a compilação de um determinado projeto, realizando a cópia de todas as bibliotecas para um determinado repositório onde poderá ser compartilhado para todos os usuários do ambiente e utilizado em outro projeto. Como estudos de caso, foram desenvolvidos dois circuitos integrados, os quais demonstraram que o ambiente virtual proposto garantiu uma maior organização, segurança, documentação e qualidade dos projetos.

Palavras-chave: Trabalho Colaborativo. Ambiente Virtual Colaborativo. Projeto em Equipe. Projeto de Circuitos Integrados em Equipe.

ABSTRACT

VIRTUAL COLLABORATIVE ENVIRONMENT FOR SUPPORT OF TEAMWORK-BASED EDUCATION OF INTEGRATED CIRCUITS

AUTHOR: Peter Prevedello

ADVISOR: André Luiz Aita

This paper describes the design and development of a virtual collaborative environment to support team-based education of integrated circuits design, named IC-COLLAB, and its implementation within the Grupo de Microeletrônica (GMICRO) of Universidade Federal de Santa Maria (UFSM). Through a simple web interface, the collaborative virtual environment provides collaborative tools that ensure project organization and efficient cooperation among participants, enhancing individual practices, teamwork, and leadership through coordination and communication. A hierarchically organized repository allows to save and share libraries developed individually by students. A version controller enables all participants to manage and document the changes made in the libraries. Communication tools allow the exchange of ideas and knowledge during the activities. Sharing of supporting materials and project suggestions can be done at any time. Finally, whenever necessary, the environment allows the coordinator to compile a given project, executing the copy of all the libraries to a particular repository where it can be shared for all the users of the environment and used in another project. As case studies, two integrated circuits were developed, which have demonstrated that the proposed virtual environment enhances the organization, safety, documentation and quality of the projects.

Keywords: Collaborative Work. Virtual Collaborative Environment. Teambased-work. Team-based-work Design of Integrated Circuits.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 2.1 – CSCW como área de pesquisa para compor um <i>Groupware</i> | 17 |
| Figura 2.2 – Modelo 3C de Colaboração e suas três dimensões: comunicação, coordenação e cooperação. | 19 |
| Figura 2.3 – Ciclo de Desenvolvimento de <i>Groupware</i> , evidenciando as cinco fases de desenvolvimento: 1-Análise de Domínio, 2- de Requisitos, 3- de Projeto, 4- de Implementação e 5- de Teste. | 25 |
| Figura 2.4 – Ambiente da <i>Cadence: Virtuoso</i> (a) <i>Schematic</i> and (b) <i>Layout Editor</i> | 27 |
| Figura 2.5 – A célula de um inversor com três visões: esquemático, símbolo e <i>layout</i> | 27 |
| Figura 2.6 – Células e bibliotecas instanciadas figuram no arquivo "cds.lib", juntamente como outras bibliotecas. | 28 |
| Figura 2.7 – Fluxo de projeto simplificado de CIs.. | 29 |
| Figura 3.1 – Componentes do IC-COLLAB classificados conforme Modelo 3C. | 36 |
| Figura 3.2 – Diagrama de caso de uso representando graficamente uma funcionalidade do IC-COLLAB. | 41 |
| Figura 3.3 – Diagrama de sequencia representando parcialmente uma sequência de processos do IC-COLLAB. | 42 |
| Figura 3.4 – Diagrama entidade-relacionamento mostrando como duas entidades se relacionam no IC-COLLAB. | 43 |
| Figura 3.5 – Arquitetura geral do IC-COLLAB. <i>Cadence</i> e IC-COLLAB estão no lado do Servidor provendo serviços aos clientes (lado dos Clientes) por meio da internet. | 44 |
| Figura 3.6 – Hierarquia de diretórios do IC-Collab. | 49 |
| Figura 3.7 – Tela inicial do IC-COLLAB. | 52 |
| Figura 3.8 – Tela de cadastro de usuários no IC-COLLAB. | 53 |
| Figura 3.9 – Passo 1 do cadastro de projeto: cadastrar informações. | 55 |
| Figura 3.10 – Passo 2 do cadastro de projeto: adicionar participantes. | 57 |
| Figura 3.11 – Passo 3 do cadastro de projeto: Definir um participante como coordenador. | 58 |
| Figura 3.12 – Passo 4 do cadastro de projeto: Criar bibliotecas para os participantes. ... | 59 |
| Figura 3.13 – Fluxograma do processo de criação de projetos no IC-COLLAB. | 60 |
| Figura 3.14 – Gerenciamento do projeto. | 61 |
| Figura 4.1 – Hierarquia de diretórios do projeto RINGOSCILATOR. | 65 |
| Figura 4.2 – Informações do projeto RINGOSCILATOR. | 66 |
| Figura 4.3 – Bibliotecas criadas no projeto RINGOSCILATOR. | 67 |
| Figura 4.4 – (a) Esquemático do Oscilador em Anel após a compilação, e (b) simulação elétrica. | 68 |
| Figura 4.5 – Informações do projeto PiezoEnergy. | 70 |
| Figura 4.6 – Bibliotecas criadas no projeto PiezoEnergy. | 71 |
| Figura 4.7 – Compartilhamento de materiais no componente <i>Wiki</i> do ambiente IC-COLLAB. | 72 |
| Figura 4.8 – Esquemático do Piezoelétrico após a compilação evidenciando os diversos blocos que modelam o sistema. | 73 |
| Figura 4.9 – Simulação elétrica do Piezoelétrico evidenciando a tensão V_{pz} sobre o piezo e as correntes no indutor LH e na bateria | 74 |
| Figura C.1 – Diagrama de Casos de Uso do IC-COLLAB. | 89 |
| Figura D.1 – Diagrama de Sequência do IC-COLLAB. | 90 |

Figura E.1 – Diagrama Entidade-Relacionamento do IC-COLLAB. 91

LISTA DE ABREVIATURAS E SIGLAS

| | |
|------------------------|---|
| <i>CAD</i> | <i>Computer-Aided Design</i> |
| <i>CI_s</i> | Circuitos Integrados |
| <i>CSCW</i> | <i>Computer Supported Cooperative Work)</i> |
| <i>DFW</i> | <i>Design Framework</i> |
| <i>GB</i> | <i>Gigabyte</i> |
| <i>GH_z</i> | <i>Gigahertz</i> |
| <i>GMICRO</i> | Grupo de Microeletrônica |
| <i>HD</i> | <i>Hard Disk</i> |
| <i>HTML</i> | <i>HyperTextMarkup Language</i> |
| <i>IC – COLLAB</i> | Ambiente Virtual Colaborativo para Apoio ao Projeto em Equipe de Circuitos Integrados |
| <i>IDE</i> | <i>Integrated Development Environment</i> |
| <i>PI</i> | Propriedade Intelectual |
| <i>PHP</i> | <i>Hypertext Preprocessor</i> |
| <i>RAM</i> | <i>Random Access Memory</i> |
| <i>SGBD</i> | Sistema de Gerenciamento de Banco de Dados |
| <i>SQL</i> | <i>Structured Query Language</i> |
| <i>TB</i> | <i>Terabyte</i> |
| <i>TIC_s</i> | Tecnologias de Informação e Comunicação |
| <i>UF_{SM}</i> | Universidade Federal de Santa Maria |
| <i>UML</i> | <i>Unified Modeling Language</i> |
| <i>WebCT</i> | <i>Web Course Tools</i> |
| <i>WWW</i> | <i>World Wide Web</i> |

SUMÁRIO

| | | |
|--------------|--|-----------|
| 1 | INTRODUÇÃO | 11 |
| 1.1 | O ENSINO COLABORATIVO | 11 |
| 1.2 | AMBIENTES VIRTUAIS DE APOIO AO TRABALHO COLABORATIVO | 12 |
| 1.3 | DESENVOLVIMENTO DE AMBIENTES VIRTUAIS | 13 |
| 1.4 | OBJETIVO DA DISSERTAÇÃO | 13 |
| 1.5 | ORGANIZAÇÃO DO TEXTO | 14 |
| 2 | FUNDAMENTAÇÃO TEÓRICA | 15 |
| 2.1 | COLABORAÇÃO E TRABALHO COLABORATIVO | 15 |
| 2.2 | AMBIENTES VIRTUAIS COLABORATIVOS | 17 |
| 2.3 | MODELO 3C DE COLABORAÇÃO | 18 |
| 2.4 | CICLO DE DESENVOLVIMENTO DE <i>GROUPWARE</i> | 22 |
| 2.5 | AMBIENTES DE CAD PARA MICROELETRÔNICA | 24 |
| 2.5.1 | <i>Cadence Design Framework</i> | 26 |
| 2.5.1.1 | <i>Desenvolvimento dos Projetos</i> | 26 |
| 2.5.1.2 | <i>Instâncias e Hierarquias</i> | 28 |
| 2.5.1.3 | <i>Fluxo de Projeto de CIs</i> | 28 |
| 2.6 | TRABALHOS RELACIONADOS | 29 |
| 2.6.1 | <i>Circuits and Systems Engineering Education Through Interdisciplinary Team-based Design Projects</i> | 30 |
| 2.6.2 | <i>A Teamwork-based Education Strategy for Teaching Lab of Analog Integrated Circuits Design</i> | 30 |
| 2.6.3 | Projeto Cooperativo no Ambiente CAVE | 31 |
| 2.6.4 | Ferramenta para gerenciamento e apoio ao desenvolvimento distribuído de <i>software</i> | 32 |
| 2.6.5 | Contribuição deste trabalho | 33 |
| 3 | IC-COLLAB: AMBIENTE VIRTUAL COLABORATIVO PARA APOIO AO ENSINO DE PROJETOS DE CIRCUITOS INTEGRADOS EM EQUIPE | 35 |
| 3.1 | MODELAGEM E PROJETO DO AMBIENTE VIRTUAL IC-COLLAB COM BASE NO CICLO DE DESENVOLVIMENTO DE <i>GROUPWARE</i> | 35 |
| 3.1.1 | Análise de Domínio | 36 |
| 3.1.2 | Análise de Requisitos | 39 |
| 3.1.3 | Projeto | 39 |
| 3.1.3.1 | <i>Diagrama de Caso de Uso</i> | 40 |
| 3.1.3.2 | <i>Diagrama de Sequência</i> | 40 |
| 3.1.3.3 | <i>Diagrama Entidade-Relacionamento</i> | 41 |
| 3.1.3.4 | <i>Arquitetura</i> | 42 |
| 3.1.3.5 | <i>Tecnologias utilizadas</i> | 43 |
| 3.1.4 | Implementação | 45 |
| 3.1.5 | Testes | 46 |
| 3.2 | VISÃO GERAL DO IC-COLLAB | 47 |
| 3.3 | DESCRIÇÃO DO FUNCIONAMENTO DO IC-COLLAB | 50 |
| 3.3.1 | Requisitos para instalação | 50 |
| 3.3.2 | Tela inicial | 51 |
| 3.3.3 | Perfis de usuário | 51 |
| 3.3.4 | Cadastro de usuários | 53 |
| 3.3.5 | Coordenador de projeto | 54 |

| | | |
|---------|---|-----------|
| 3.3.6 | Cadastro de projetos | 54 |
| 3.3.7 | Gerenciamento do projeto | 58 |
| 4 | ESTUDOS DE CASO | 64 |
| 4.1 | ESTUDO DE CASO 1 | 64 |
| 4.1.1 | Cadastro de Usuários | 64 |
| 4.1.2 | Cadastro do projeto | 64 |
| 4.1.3 | Projeto Colaborativo | 67 |
| 4.1.3.1 | <i>Compilação e simulação</i> | 68 |
| 4.2 | ESTUDO DE CASO 2 | 69 |
| 4.2.1 | Cadastro do projeto | 69 |
| 4.2.2 | Projeto Colaborativo | 69 |
| 4.2.2.1 | <i>Compilação e simulação</i> | 70 |
| 5 | CONCLUSÃO | 75 |
| 5.1 | CONTRIBUIÇÕES | 76 |
| 5.2 | TRABALHOS FUTUROS E PERSPECTIVAS | 77 |
| | REFERÊNCIAS BIBLIOGRÁFICAS | 79 |
| | ANEXO A – REQUISITOS FUNCIONAIS (RF) | 84 |
| | ANEXO B – REQUISITOS NÃO-FUNCIONAIS (RNF) | 88 |
| | ANEXO C – DIAGRAMA DE CASO DE USO | 89 |
| | ANEXO D – DIAGRAMA DE SEQUÊNCIA | 90 |
| | ANEXO E – DIAGRAMA ENTIDADE-RELACIONAMENTO | 91 |
| | ANEXO F – ARTIGO PUBLICADO NO ISCAS 2018 | 92 |

1 INTRODUÇÃO

Na sociedade atual observa-se uma crescente necessidade de se realizar tarefas e solucionar problemas de modo colaborativo, ou seja, de se interagir e trocar conhecimentos por meio da comunicação para atingir metas e objetivos compartilhados. No desenvolvimento de projetos ou no ensino e aprendizagem colaborativos, seja na indústria ou no meio acadêmico, a interação entre os seus membros é de enorme importância. Isto porque as habilidades individuais precisam ser combinadas com as de outras pessoas, para que se produza um trabalho de maior qualidade. Assim também como o trabalho combinado em um projeto onde atuam diversos colaboradores que, compartilhando ideias e conhecimento sob uma boa coordenação, pode prover a complementação de capacidades e gerar melhor resultado do que se fosse realizado individualmente (FUKS; RAPOSO; GEROSA, 2002).

1.1 O ENSINO COLABORATIVO

O ensino colaborativo na academia mostra-se crescente também. Ainda que não seja regra, a literatura tem mostrado um número crescente de casos onde o ensino, no caso desse trabalho projeto de CIs, tem ocorrido seguindo uma abordagem baseada em equipes. Por exemplo, em Dehaene et al. (2011), os autores relatam sua convicção de que a forma mais apropriada de ensino e aprendizagem é através de projetos por experiência prática, realizada por grupos de alunos. Uma abordagem semelhante é relatada em Rosa (2012), onde é proposta uma metodologia baseada em equipe para trabalhar o conteúdo teórico e prático de projetos de CI analógicos. De acordo com o texto, os alunos são encorajados a participar ativamente do curso trabalhando em conjunto com o objetivo comum de projetar circuitos. Também em Fernandes (2016), o autor sugere um modelo pedagógico escolhido para enfatizar o projeto prático de um CI que cumpra um conjunto de especificações. Enquanto o texto se concentra no conteúdo do programa de um curso de projeto de CIs, o método de ensino segue uma abordagem baseada em projetos que está alinhado com os trabalhos mencionados anteriormente.

Sendo assim, observou-se que a metodologia de ensino e aprendizagem colaborativa aplicada em projetos em equipe apresentou diversas vantagens, como:

- maior motivação dos alunos quando o projeto é realizado em equipe;
- melhoria da aprendizagem dos alunos através do compartilhamento de ideias e conhecimento;
- habilidades técnicas e complementares como liderança, tomada de decisão e a capacidade de aplicar a teoria em situações práticas são adquiridas mais facilmente;
- atividades práticas em equipe envolvendo situações reais de projeto de CIs preparam

os alunos para o mercado profissional, onde desenvolvem e assumem responsabilidades específicas durante o trabalho.

Desta forma, além das vantagens observadas, o ensino e aprendizagem colaborativo baseado em projetos demonstrou-se também como uma metodologia de ensino alternativa para ser aplicada em disciplinas práticas dos cursos de engenharia.

Porém, a complexidade de se estruturar um curso para seguir uma abordagem de ensino baseada em equipe requer ferramentas computacionais que ofereçam meios de interação e colaboração para apoiar este processo. Ferramentas que permitam disponibilizar materiais didáticos, definir um conjunto de tarefas para uma determinada equipe, espaço virtual organizado e compartilhado, bem como monitorar as atividades individuais dos alunos são requisitos mínimos necessários para o professor estruturar um trabalho colaborativo envolvendo um projeto de CI.

1.2 AMBIENTES VIRTUAIS DE APOIO AO TRABALHO COLABORATIVO

No final da década de 1990, inicia-se a utilização da *Internet* como apoio virtual ao trabalho do professor nos mais diferentes níveis de ensino. Neste período surgiram projetos que envolviam o desenvolvimento de *softwares* educativos e de ambientes virtuais de ensino e aprendizagem, estimulando o uso da tecnologia e da interação entre alunos e professores (FAGUNDES, 1996).

Atualmente, muitos desses ambientes virtuais buscam evoluir de forma a suprir algumas soluções para os problemas frequentes no ensino como a motivação dos alunos, a liderança e a relação entre teoria e prática. Nestes ambientes virtuais foram inseridas ferramentas de interação entre os usuários, como por exemplo, *chat*, fórum, agendas, compartilhamento de documentos, bem como espaços onde os usuários podem construir seus trabalhos de modo colaborativo.

Dentro deste contexto, podemos definir um ambiente virtual de ensino como um conjunto de componentes de comunicação capazes de oferecer suporte à colaboração em ambientes educacionais permitindo que haja interação entre pessoas localizadas em espaços físicos eventualmente diferentes e possibilitando a troca de ideias, informações e conhecimentos de forma rápida e eficiente (BRITO; PEREIRA, 2004)(SCHAF, 2011)(FUKE; PIMENTEL, 2011).

De outra forma, ambientes virtuais possuem características que suprem os requisitos necessários para se estruturar um curso seguindo uma abordagem de ensino baseada em equipe.

Entretanto, o processo de desenvolvimento de ambientes virtuais colaborativos requer além de habilidades técnicas de desenvolvimento de *software*, sendo necessário a compreensão do processo de colaboração. Tal compreensão revela-se fundamental,

baseia-se em aspectos técnicos mas não obstante depende também de experiências pessoais (e da própria equipe) no assunto onde se colabora.

1.3 DESENVOLVIMENTO DE AMBIENTES VIRTUAIS

Os métodos da engenharia de *software* que muito evoluíram no desenvolvimento de sistemas não se mostraram suficientes para impulsionar o desenvolvimento de ambientes virtuais. A complexidade de se desenvolver um ambiente virtual existe devido às mudanças nas características das equipes de trabalho ao longo do tempo. Mesmo que um programador consiga desenvolver um ambiente satisfatório para uma determinada equipe, com o tempo aparecem novas situações e problemas que o tornam inadequado (BRITO; PEREIRA, 2004).

Deste modo, apesar dos ambientes colaborativos serem bastante utilizados para auxiliar equipes de trabalho de diversas áreas (MARÓTI et al., 2014) (SAWICKI; INDRUSIAK; REIS, 2002), algumas dessas áreas exigem componentes com funcionalidades específicas, tornando difícil o seu desenvolvimento e administração, o que requer metodologias específicas para sua construção.

A literatura apresenta alguns modelos e métodos específicos para o desenvolvimento de ambientes virtuais (DEWAN, 2001)(FISCHER et al., 2001)(LUKOSCH; SCHÜMMER, 2006)(PIMENTEL et al., 2006)(FUKS et al., 2003). Os dois principais são: o Modelo 3C de Colaboração (PIMENTEL et al., 2006) e o Ciclo de Desenvolvimento de *Groupware* (FUKS et al., 2003). Enquanto o modelo 3C de Colaboração analisa a colaboração sob três dimensões, a saber comunicação, coordenação e cooperação, o Ciclo de Desenvolvimento de *Groupware* define uma estratégia para construção de ambientes virtuais baseadas em fases.

1.4 OBJETIVO DA DISSERTAÇÃO

Considerando o exposto, a necessidade de se trabalhar em equipe durante a execução de circuitos integrados complexos, e de um ambiente que permita a colaboração, este trabalho tem como objetivo apresentar o desenvolvimento e implantação de um ambiente virtual colaborativo para apoiar o ensino de CIs baseado em equipe no Grupo de Microeletrônica (GMICRO) da Universidade Federal de Santa Maria (UFSM).

O ambiente virtual fornece ferramentas de colaboração, garantindo a organização de projetos através da coordenação e uma cooperação eficiente entre os participantes (suportadas por ferramentas de comunicação), aprimorando não só as práticas individuais, mas também o trabalho em equipe e a liderança. Um repositório principal organizado permite salvar e compartilhar bibliotecas individualmente desenvolvidas pelos alunos. Usando um controlador de versão, cada participante é capaz de rastrear e gerenciar mudanças

em suas bibliotecas. Materiais de suporte e tutoriais podem ser criados e postados para os participantes. E, após a conclusão do projeto, mantém as bibliotecas armazenadas e documentadas para outros usuários ou cursos do grupo, como repositórios locais de Propriedade Intelectual (PI).

1.5 ORGANIZAÇÃO DO TEXTO

Este trabalho está estruturado como segue.

No capítulo dois são apresentados os fundamentos do trabalho colaborativo e de ambientes virtuais colaborativos. Modelos de colaboração e de desenvolvimento específicos como o Modelo 3C de Colaboração, e o Ciclo de Desenvolvimento de *Groupware* são também visitados. O ambiente *Cadence Design Framework* e o fluxo de projetos de CIs são também explorados. Por fim, trabalhos relacionados ao tema são apresentados.

O capítulo três descreve o processo de desenvolvimento e implantação do ambiente virtual, apresentando primeiramente o processo de análise, modelagem e implementação através de fases do Ciclo de Desenvolvimento de *Groupware*. Em seguida é apresentada uma visão geral do sistema e por fim suas funcionalidades são detalhadas.

No capítulo quatro são apresentados os estudos de caso, que ajudaram, não somente a identificar as falhas do sistema, mas também a aprimorar as funcionalidades para que o sistema tivesse suas funcionalidades ajustadas e melhoradas.

Por fim, o capítulo seis apresenta as conclusões do trabalho com os resultados coletados através dos estudos de caso e suas contribuições, bem como os trabalhos futuros e perspectivas.

2 FUNDAMENTAÇÃO TEÓRICA

Nesse capítulo os fundamentos do trabalho colaborativo serão abordados para nortear o desenvolvimento de ambientes virtuais colaborativos. O propósito é identificar as premissas mais elementares quando se desenvolve um trabalho em grupo de forma organizada. Primeiramente, são identificados os principais conceitos sobre o trabalho colaborativo e ambientes virtuais colaborativo. Com base nisso, o Modelo 3C de Colaboração é apresentado como meio de classificação de ferramentas para compor um ambiente colaborativo. A seguir, métodos para construção de ambientes colaborativos, e tecnologias necessárias serão abordadas. Em seguida, será apresentada a ferramenta *Cadence*, demonstrando seu processo de construção de CIs, os quais necessitam ser organizados. Por fim, alguns trabalhos da literatura relacionados ao presente foram discutidos.

2.1 COLABORAÇÃO E TRABALHO COLABORATIVO

O termo colaboração, antes de mais nada, precisa ser contextualizado para definir a relação desejada entre os participantes (BRNA, 1998). A colaboração é uma situação em que duas ou mais pessoas aprendem e cooperam sobre algo em comum (DILLENBOURG, 1999). Mais especificamente, a colaboração baseia-se no modelo onde o conhecimento e a cooperação podem ser desenvolvidos dentro de uma população onde membros interagem ativamente compartilhando experiências e assumindo papéis diferentes, produzindo melhores resultados do que se atuassem individualmente (FUKS; RAPOSO; GEROSA, 2002). Além disso, Fuks, Raposo e Gerosa (2002) complementam afirmando que a colaboração entre membros da equipe pode prover a complementação de capacidades, conhecimentos individuais e também minimizar inconsistências e falhas. De acordo com Barros (1994):

colaborar (co-labore) significa trabalhar junto, que implica no conceito de objetivos compartilhados e uma intenção explícita de somar algo – criar alguma coisa nova ou diferente através da colaboração, contrapondo-se a uma simples troca de informação ou de instruções.

A compreensão de que outras pessoas agregam valor ao trabalho é um grande motivador para colaboração. Quando uma tarefa é complexa e/ou requer múltiplas competências para a sua execução, é necessário unir os esforços de várias pessoas. Essa união de pessoas define uma equipe de trabalho. Vários elementos estão envolvidos na formação de uma equipe, como por exemplo, participantes, motivação, compromisso, objetivos, etc (FUKS; PIMENTEL, 2011). Desse modo, além dos elementos mencionados, o trabalho colaborativo requer que os indivíduos possuam habilidades para realizar tarefas em equipe de maneira organizada, comunicativa e com bom relacionamento (Morandini,

Marcelo, 2017). Em seu trabalho, Fuks e Pimentel (2011) argumentam que o trabalho colaborativo não requer apenas motivação dos participantes. Observe que, se os objetivos não estiverem alinhados e organizados, os participantes trabalharão em sentidos variados, e mesmo em oposição, o que irá prejudicar sensivelmente o resultado final.

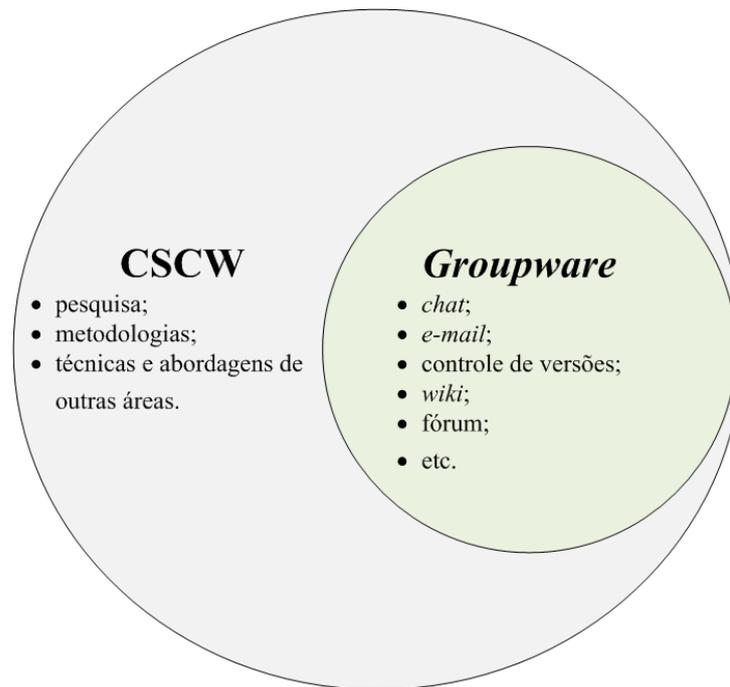
Em seu trabalho Grosz (1996) exemplifica a colaboração na preparação de um jantar por uma equipe. O objetivo é comum, de modo que se um falhar, o jantar falha. Desta forma, eles são obrigados a se ajudar. Quando um for ao supermercado, deve trazer os ingredientes para os outros, e todos negociam o cardápio, para que seja coerente, e o uso dos recursos, para que um não utilize um recurso imprescindível para outro. A comunicação é voltada para ação, com objetivo de negociar e trocar informações. A coordenação é feita em duas etapas: na preparação para o trabalho e dinamicamente enquanto ele acontece. A cooperação acontece na realização conjunta das tarefas. Os indivíduos planejam juntos, atuam conjuntamente, negociam e possuem um objetivo compartilhado, sendo que a falha de um implica na falha de todos.

No contexto do trabalho colaborativo, surgiu a área de pesquisa, denominada *Computer Supported Cooperative Work* (CSCW), que estuda e utiliza métodos, técnicas e abordagens de diferentes áreas, como psicologia, educação, ciência da computação, sistemas distribuídos, entre outras, e fornecer conhecimento para o desenvolvimento de tecnologias computacionais que ofereçam suporte ao trabalho colaborativo. As tecnologias computacionais, por sua vez, são ambientes virtuais denominados *groupware*. Esses sistemas computacionais, denominados *Groupware*, devem ser especializados o bastante, a fim de oferecer aos seus usuários formas de interação facilitando o controle, a coordenação, a cooperação e a comunicação entre as partes envolvidas que compõem a equipe, tanto no mesmo local, como em locais geograficamente diferentes e que as formas de interação aconteçam tanto ao mesmo tempo ou em tempos diferentes (PIMENTEL; FUKS; LUCENA, 2008). Especificamente, definimos *groupware* como um ambiente colaborativo baseado em computador (virtual) para dar suporte a grupos de pessoas engajadas numa tarefa (ou objetivo) em comum e que provê uma interface para um ambiente compartilhado e organizado.

Conforme a Figura 2.1, CSCW e *groupware* são dois termos distintos, pois enquanto CSCW, em um contexto mais amplo, é usado para designar a pesquisa na área do trabalho em grupo e como os computadores podem apoiá-lo, o *groupware* é usado para designar a tecnologia (*hardware* e /ou *software*) gerada pela pesquisa em CSCW (BOTELHO; VIDAL, 2005). No entanto, a ideia inerente a ambos é auxiliar o trabalho colaborativo.

Visando ficar em linha com o título deste trabalho, é importante destacar que o termo *groupware* é equivalente a ambiente virtual colaborativo.

Figura 2.1 – CSCW como área de pesquisa para compor um *Groupware*



Fonte: Próprio autor.

2.2 AMBIENTES VIRTUAIS COLABORATIVOS

A necessidade de colaboração não ocorre apenas no cumprimento de tarefas, mas também nos processos de ensino e aprendizagem. Desta maneira, em algum ponto, a interação e colaboração serão necessárias de modo que o conhecimento buscado seja realmente assimilado ou construído (FUKS; PIMENTEL, 2011). Quando o ensino e a aprendizagem são propostos através de trabalhos acadêmicos individuais, normalmente há a necessidade apenas de uma ferramenta com elementos que possibilitem a organização do conteúdo dos trabalhos em desenvolvimento. Porém, quando o ensino ocorre através de trabalhos em equipe que exigem colaboração entre os participantes, é requerido uma ferramenta ou ambiente com elementos que ofereçam muito mais que a simples organização. É necessário um ambiente que suporte a comunicação e cooperação entre os participantes e coordenação destes, para que as atividades sejam concluídas com sucesso, e os conhecimentos advindos do trabalho em equipe sejam consolidados e aprimorados (através da interação) (FILHO et al., 2005).

Assim, os sistemas ambientes virtuais colaborativos atuam para permitir que materiais didáticos sejam oferecidos e que ferramentas colaborativas sejam usadas de modo que o trabalho em equipe ocorra de forma independente de tempo e espaço (BRITO; PEREIRA, 2004). Um ambiente virtual é um "espaço físico" simulado através de tecnologias computacionais onde os participantes não necessitem estar fisicamente juntos para interagir e compartilhar informações em tempo real (SCHAF, 2011)(FUKS; PIMENTEL, 2011). Os ambientes virtuais historicamente têm sido aplicados nas diversas áreas do conhe-

cimento, como por exemplo: simuladores, jogos de entretenimento, jogos educacionais, treinamentos a distância, educação a distância, etc. A ideia central é reproduzir o que existe no mundo real e fornecer uma interação mais rica entre os integrantes (SNOWDON; CHURCHILL; MUNRO, 2001).

Bastante utilizados no ensino e na aprendizagem à distância ou semi-presencial, os ambientes virtuais também começaram a ser desenvolvidos para apoiar equipes de trabalho, distribuídas geograficamente, com necessidades de interação e troca de informações de modo organizada. Essa interação é possível através de uma interface digital que faz a mediação entre o usuário e as ferramentas que compõem o ambiente. Cada ferramenta possui características para resolver certas questões durante a colaboração, por exemplo, uma reunião virtual através de um *chat*, que integrada a outras ferramentas com características diferentes, complementam-se para que os objetivos da equipe possam ser alcançados. Ferramentas como: *chat*, *e-mail*, gerenciador de repositório, *wiki*, fórum, controlador de versões, entre outras, quando integradas possuem características funcionais para trabalhos em equipe e normalmente são as mais aplicáveis em ambientes virtuais colaborativos.

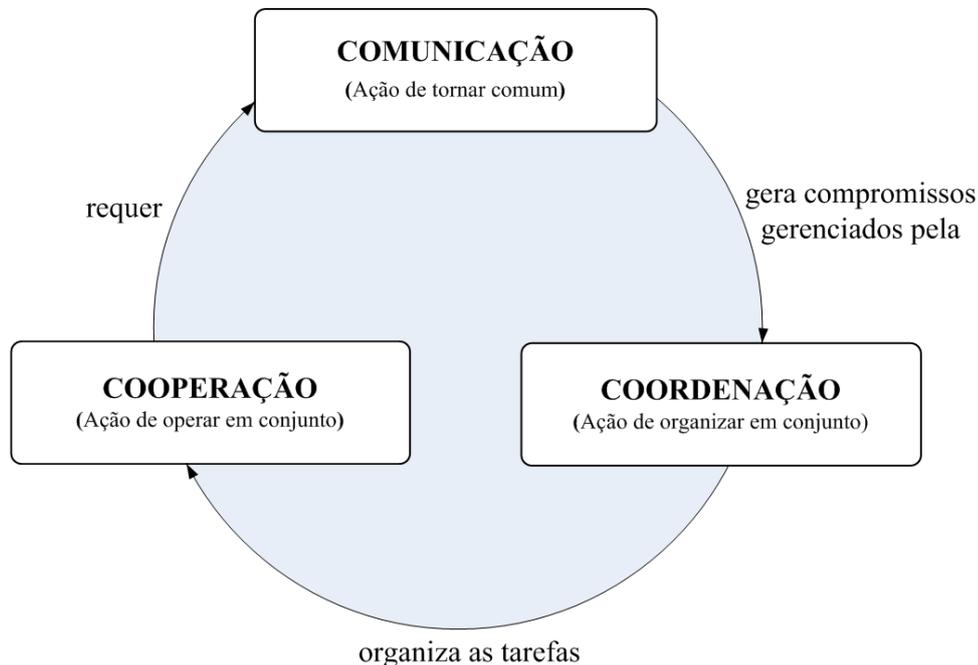
Segundo Gerosa (2006), o desenvolvimento de ambientes virtuais colaborativos requer uma profunda compreensão do trabalho colaborativo e de suas necessidades fundamentais para operacionalização. Nessa busca por uma melhor compreensão das necessidades dos ambientes colaborativos, modelos de cooperação foram propostos. Entre esses, um dos padrões aplicáveis para se analisar e desenvolver ambientes virtuais colaborativos é o modelo baseado em três dimensões, a saber: comunicação, cooperação e coordenação, daí a denominação de Modelo 3C, visto a seguir. Nesse modelo, os três elementos se mostraram essenciais para o desenvolvimento e implementação de um ambiente virtual para o trabalho colaborativo.

2.3 MODELO 3C DE COLABORAÇÃO

Ellis, Gibbs e Rein (1991) classificaram os sistemas que dão suporte ao trabalho em equipe em três dimensões: comunicação, coordenação e colaboração. Essa classificação deu origem ao Modelo 3C de Colaboração, formulado posteriormente. Nesse modelo, a cooperação designa estritamente a ação de operar em conjunto, enquanto a colaboração designa a ação de realizar todo o trabalho em conjunto, o que envolve comunicação, coordenação e cooperação (FUKS; PIMENTEL, 2011).

O Modelo 3C de Colaboração é frequentemente usado pela literatura para classificar os *groupwares*, tal como proposto por Schlichter e Borghoff (2000) e Laurillau e Nigay (2002). Em Gerosa (2006), além de ser utilizado para classificar, o Modelo 3C é utilizado como base para a modelagem e desenvolvimento de *groupware*, onde cada C é analisado detalhadamente. Já em Fuks et al. (2003), os autores propõem um método de desenvol-

Figura 2.2 – Modelo 3C de Colaboração e suas três dimensões: comunicação, coordenação e cooperação.



Fonte: Adaptado de Fuks et al. (2003).

vimento de *groupware* baseado no Modelo 3C de Colaboração, chamado de "Engenharia de *Groupware*".

No Modelo 3C, esquematizado na Figura 2.2, a separação em dimensões foca nos aspectos mais relevantes para a análise da colaboração, entretanto, os C's se inter-relacionam para que a colaboração ocorra. As trocas ocorridas durante a comunicação geram compromissos que são gerenciados pela coordenação, que por sua vez organiza e distribui as tarefas que são executadas na cooperação. Ao cooperar os indivíduos têm necessidade de se comunicar para negociar (e renegociar) e para tomar decisões sobre situações não previstas inicialmente. Isto mostra o aspecto cíclico da colaboração. Através da percepção, o indivíduo se informa sobre o que está acontecendo, sobre o que as outras pessoas estão fazendo e coleta informações necessárias para seu trabalho.

Sob essa perspectiva, um *groupware* deve em essência dar suporte aos três C's. Ainda que o objetivo de uma ferramenta possa voltar-se para o suporte específico de um dos C's, ainda assim contemplará aspectos dos demais C's. Por exemplo, um sistema de mensagem como o *e-mail*, embora pensado e projetado para estabelecer a comunicação, também pode ser usado para distribuir ordens aos participantes, ou coordenar o trabalho de uma equipe, servindo também de suporte à coordenação (PIMENTEL et al., 2006). A seguir, as dimensões do Modelo 3C de Colaboração e suas inter-relações são detalhadas:

- Comunicação

No trabalho em equipe, a comunicação é voltada para a ação. Esta comunicação

normalmente envolve negociação e tomada de decisões que geram tarefas. A necessidade de renegociar e tomar decisões sobre situações imprevistas que ocorrem durante a cooperação demanda comunicação que, por sua vez, demanda coordenação para organizar as tarefas (FUKS; PIMENTEL, 2011). Uma ferramenta de comunicação mediada por computador dá suporte às interações entre os participantes, podendo gerenciar a troca de informações e armazená-las mantendo um histórico para posterior consulta (FUKS et al., 2003). A comunicação pode ser classificada em síncrona ou assíncrona, detalhadas a seguir.

1. Comunicação síncrona

Na comunicação síncrona, o emissor e o receptor devem estar num estado de sincronia antes da comunicação iniciar e permanecer em sincronia durante a transmissão, ou seja, ambos devem estar simultaneamente conectados ao *groupware* para que a colaboração se efetive (FILIPPO et al., 2007). Em ferramentas de comunicação síncrona, valoriza-se a velocidade da interação, visto que o tempo de resposta entre a ação de um participante e a reação dos demais é curto (FUKS et al., 2003). O exemplo clássico é a comunicação por troca de mensagens através de *chat*.

2. Comunicação assíncrona

Em uma comunicação assíncrona, o emissor envia uma mensagem ao receptor e esse não necessariamente a recebe no mesmo momento, ou se a recebe, não necessariamente a acessa naquele mesmo momento, ou seja, eles não precisam estar simultaneamente conectados ao *groupware* (FILIPPO et al., 2007). Ferramentas de comunicação assíncrona normalmente são utilizadas quando se deseja valorizar a reflexão dos participantes, pois estes terão mais tempo antes de agir (FUKS et al., 2003). Um exemplo de comunicação assíncrona é o *e-mail*.

• Coordenação

Para garantir o cumprimento das tarefas firmadas nas decisões durante a comunicação e a realização do trabalho colaborativo através da soma dos trabalhos individuais, é necessária a coordenação das atividades para se atingir os objetos (FUKS; PIMENTEL, 2011). O coordenador organiza a equipe para evitar que esforços de comunicação e de cooperação sejam perdidos e para que as tarefas sejam realizadas na ordem correta, no tempo definido e cumprindo as restrições e objetivos (RAPOSO et al., 2004). Sem coordenação, há o risco de os participantes se envolverem em tarefas conflitantes ou repetitivas, ou mesmo até em tarefas desnecessárias ou que não agregam ao objetivo final.

A coordenação envolve a pré-articulação das tarefas, o gerenciamento do andamento das mesmas e a pós-articulação. A pré-articulação envolve as ações necessárias para preparar a colaboração, normalmente concluídas antes do trabalho colaborativo se iniciar: identificação dos objetivos, mapeamento destes objetivos em tarefas, seleção dos participantes, distribuição das tarefas entre eles, etc. A pós-articulação ocorre após o término das tarefas, e envolve a avaliação e análise das tarefas realizadas e a documentação do processo de colaboração (FUKS et al., 2003).

As tarefas da colaboração são originadas dos compromissos assumidos na comunicação. O grupo se coordena através de mecanismos de comunicação (ferramentas como o *chat* e *e-mail*) de forma a garantir a execução das tarefas, respeitando suas interdependências (PIMENTEL et al., 2006).

- **Cooperação**

Comunicação e coordenação, apesar de vitais, não são suficientes por si só. É necessário um espaço compartilhado, ainda que virtual, para que todos os membros da equipe possam trabalhar em conjunto (FUKS; PIMENTEL, 2011). Cooperação é a operação conjunta dos participantes da equipe em um espaço compartilhado visando a realização das tarefas gerenciadas pela coordenação. Indivíduos cooperam produzindo, manipulando e organizando informações, construindo e refinando objetos de cooperação, como documentos, planilhas, gráficos, etc (FUKS et al., 2003).

O produto ou trabalho final da cooperação é um ou mais objetos desenvolvidos pelos participantes. Muitas vezes, eles preferem manter versões privadas dos objetos, pois assim têm mais liberdade para trabalhar antes de compartilhar com os demais participantes. Porém, sem a coordenação apropriada, esta forma de trabalhar gera inconsistências no artefato compartilhado e requerer mais trabalho para integrar as partes desenvolvidas individualmente (FUKS; RAPOSO; GEROSA, 2002).

Um *groupware* geralmente integra um conjunto de ferramentas para colaboração, que, de alguma forma implementam os componentes 3C. Em seu trabalho, Pimentel et al. (2006) fez um comparativo listando diferentes ferramentas disponíveis em diferentes sistemas colaborativos para descobrir quais as ferramentas mais ofertadas, e constatou que a maioria dos sistemas analisados oferece fórum, *chat*, *e-mail*, agenda, relatórios de atividades, questionários, gerenciamento de tarefas e gerenciamento de arquivos. Cada ferramenta pode ser vista de forma relativamente autônoma dentro de um *groupware*. Estas características são propícias à aplicação de técnicas de desenvolvimento baseadas em componentes, onde as ferramentas são componentes do sistema colaborativo a serem instanciados e configurados (GEROSA, 2006).

2.4 CICLO DE DESENVOLVIMENTO DE *GROUPWARE*

Todo o ambiente virtual, seja colaborativo ou não, possui uma equipe de pessoas que o desenvolveu e o administra. Durante o desenvolvimento, essa equipe preocupa-se com diversos fatores que qualificam o ambiente, como: conteúdos, interfaces, ferramentas e desempenho. É certo que a qualidade de um ambiente está relacionada com a compreensão e consequente capacidade de modificações evolutivas que seus desenvolvedores realizam. Sob o ponto de vista da engenharia de *software*, devem ser aplicadas algumas técnicas de desenvolvimento de *software* combinadas com modelos conceituais e aspectos humanos originados do CSCW. Essa combinação descreve um modelo de engenharia de *groupware*, que segundo Fuks, Gerosa e Lucena (2002), é um processo sistemático, disciplinado e instrumentado pelo qual se modela e se desenvolve um ambiente virtual colaborativo.

Proposto por Fuks et al. (2003), o Ciclo de Desenvolvimento de *Groupware* foi elaborado para o desenvolvimento de ambientes virtuais colaborativos com base no Modelo 3C de Colaboração. O ciclo é baseado no modelo espiral da engenharia de *software*, conforme mostrado na Figura 2.3, e possui as seguintes fases de desenvolvimento:

1. Análise de Domínio

A fase de Análise de Domínio é o processo de identificar, coletar, analisar e organizar informações sobre as aplicações já existentes. O Modelo 3C instrumenta o desenvolvedor para analisar as aplicações com base nos conceitos referentes as dimensões comunicação, coordenação e cooperação. Considere o exemplo de um sistema de mensagem assíncrona como o *e-mail*, ainda que mais caracterizado pela comunicação, também abstrai as dimensões coordenação e cooperação.

A fase de Análise do Domínio é utilizada como base para a definição e a especificação dos requisitos e é onde são identificados e organizados os conhecimentos sobre um conjunto de problemas a serem solucionados.

2. Análise de Requisitos

Requisitos funcionais e não funcionais de comunicação, coordenação e cooperação são levantados na fase de Análise de Requisitos. O requisito é dito funcional quando descreve uma função do sistema ou serviço que o sistema deve realizar, como por exemplo, o cadastro de usuários. Já os requisitos não-funcionais, normalmente em menor número, são aqueles relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, e tecnologias envolvidas, como por exemplo, a arquitetura sobre a qual o sistema será construído (PRESMAN, 2011).

Esses requisitos muitas vezes não são claros ou suficientes para uma especificação precisa da aplicação, porém são muito importantes para o desenvolvimento do

ambiente como um todo (GUTWIN; GREENBERG, 2000).

Segundo Sommerville (2011), o processo de levantamento e análise de requisitos contém as seguintes atividades:

- Compreensão do domínio:
Os analistas devem desenvolver sua compreensão do domínio da aplicação;
- Coleta de requisitos:
É o processo de interagir com os envolvidos (alunos e professores) do sistema para descobrir seus requisitos. A compreensão do domínio se desenvolve mais durante essa atividade;
- Verificação de requisitos:
Os requisitos são verificados para descobrir se estão completos e consistentes e se estão em concordância com o que os envolvidos desejam do sistema.

É importante que os requisitos sejam definidos e documentados de maneira que possam ser utilizados na próxima fase do ciclo (fase de Projeto) a fim de verificar e compreender as dificuldades e viabilidade do projeto que se pretende implementar com base nas funcionalidades levantadas (requisitos) (SOMMERVILLE, 2011).

O desenvolvimento em espiral possibilita avaliar e validar continuamente os requisitos, bem como as decisões de projeto e de implementação.

3. Análise de Projeto

Na fase de Projeto, os processos definidos através dos requisitos devem ser mapeados e documentados através de diagramas de processos. Nesse processo de modelagem, faz-se necessária uma linguagem comum entre os envolvidos (e.g. desenvolvedor e cliente) para que eles possam documentar e debater sobre o projeto do sistema em questão. Diversas linguagens e extensões podem ser utilizadas para a modelagem de *software*, dentre elas a UML (*Unified Modeling Language*), que permite representar um sistema de forma padronizada (com intuito de facilitar a compreensão pré-implementação) a qual possibilita representar objetos específicos de *software* e da colaboração através de diagramas (BOOCH; RUMBAUGH; JACOBSON, 2006) (PRESMAN, 2011).

4. Implementação

A fase de Implementação consiste em traduzir tudo o que foi analisado, projetado e documentado anteriormente para uma forma passível de execução pela máquina. Esta tradução pode ser descrita na forma de algoritmos e estruturas de dados e, portanto, deve ser codificada em uma linguagem de programação (GEROSA, 2006).

Esta atividade é também chamada de codificação de *software* utilizando ferramentas que auxiliam no processo, como por exemplo, IDEs (*Integrated Development Environment*) e *frameworks* (PRESMAN, 2011).

A fase de Implementação constitui no desenvolvimento de uma versão do ambiente virtual colaborativo, que posteriormente será disponibilizado à fase de Testes.

5. Teste

Por fim, na fase de Testes, testes e avaliações utilizando dados fictícios serão realizadas de modo a refinar o ambiente virtual iterativamente. Nas empresas fabricantes de *software*, normalmente os testes e validações de sistemas são realizados por pessoas destinadas a realizar somente essa função, o que maximiza a o processo e qualifica o sistema (PRESMAN, 2011).

Segundo Sommerville (2011), a fase de Testes tem dois objetivos distintos:

- Teste de defeitos

Descobrir situações em que o sistema apresenta erros/falhas, ou seja, identificar comportamentos indesejáveis do sistema, tais como pane, processamentos incorretos e dados corrompidos;

- Teste de validação

Demonstrar ao cliente que o sistema atende aos requisitos. Para tal, deve-se realizar pelo menos um teste para cada requisito documentado, nos quais se espera que o sistema execute corretamente o que foi requisitado pelo cliente.

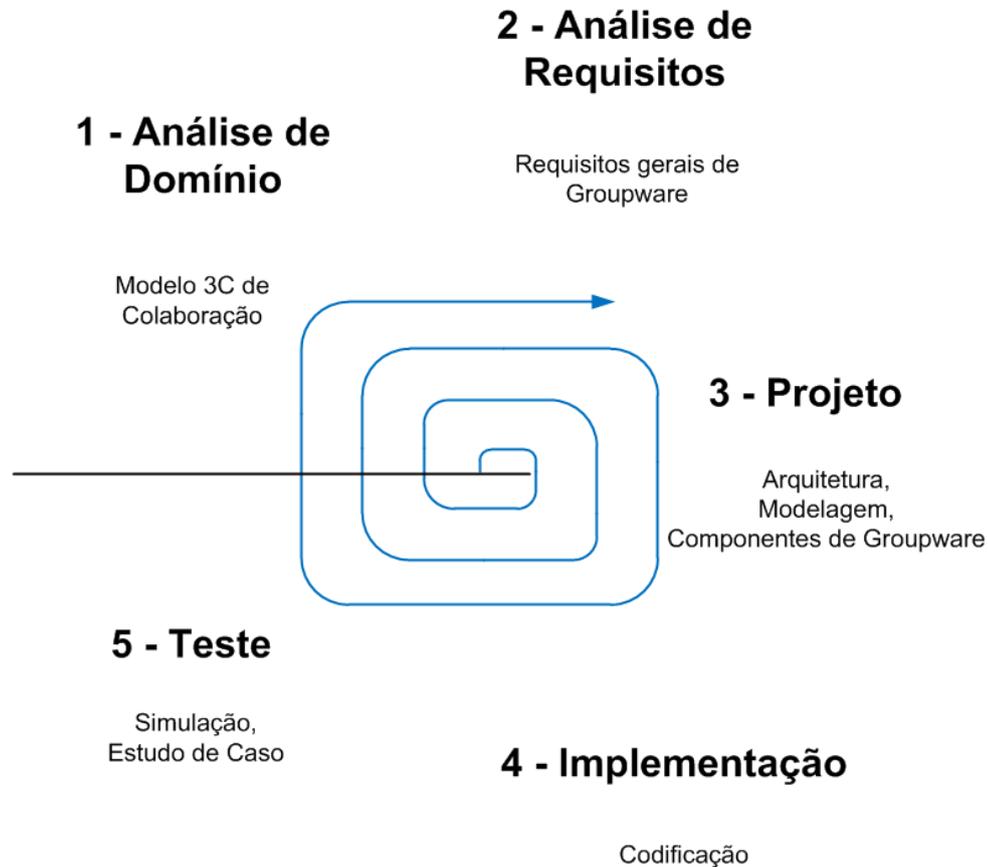
Caso seja necessário modificar a implementação de um componente ou até mesmo desenvolver um novo, o ciclo será reiniciado até que o sistema esteja em condições ideais ou satisfatórias para utilização.

Dado a característica espiral, este modelo tem a vantagem de combinar o modelo sequencial com a interatividade da prototipação, muito necessária no desenvolvimento de um *groupware*. Prototipação é fundamental para a construção de um *groupware*, dado sua tendência a falhas. O desenvolvimento através do modelo espiral com sucessivos testes de aceitação e de usabilidade adapta o *groupware* às reais necessidades dos usuários.

2.5 AMBIENTES DE CAD PARA MICROELETRÔNICA

Ambientes de CAD para microeletrônica, de maneira geral, disponibilizam uma série de ferramentas para o projeto de circuitos integrados (CIs). Como objetivo principal, esses ambientes permitem que projetos completos sejam desenvolvidos sem a necessidade de se utilizar outras ferramentas e/ou plataformas, o que repercute diretamente na velocidade

Figura 2.3 – Ciclo de Desenvolvimento de *Groupware*, evidenciando as cinco fases de desenvolvimento: 1-Análise de Domínio, 2- de Requisitos, 3- de Projeto, 4- de Implementação e 5- de Teste.



Fonte: Adaptado de Fuks et al. (2003).

com que CIs são projetados. A evolução das tecnologias, contudo, requer constante atualização de ferramentas (e criação de outras novas) que facilitem o projeto e desenvolvimento de CIs (MACMILLEN et al., 2000).

Dentre as diversas soluções propostas para lidar com a diversidade de áreas envolvidas no projeto de CIs, duas categorias de software merecem destaque: *frameworks* e as ferramentas independentes. *Frameworks* de grande porte apresentam soluções completas para o projeto de CIs, enquanto ferramentas independentes lidam com tarefas específicas. Um *framework* é de fato mais que um simples conjunto de ferramentas, pois ele provê diversas funcionalidades para auxiliar o fluxo de trabalho dos usuários (HARRISON et al., 1990). Ferramentas independentes, por outro lado, apresentam usualmente desempenho superior na tarefa que executam em função de sua especificidade.

Entre os diversos ambientes de CAD para microeletrônica, pode-se citar o ambiente de trabalho *Cadence Design Framework*. O foco nesse ambiente de projeto advém do fato que o mesmo, no momento, vem sendo utilizado em atividades de ensino, pesquisa e também no desenvolvimentos de trabalhos científicos nas áreas de engenharia elétrica e de computação da Universidade Federal de Santa Maria (UFSM). Convém ressaltar que

não é escopo desse trabalho detalhar o ambiente *Cadence*, mas descrever características que merecem atenção sob a óptica do trabalho colaborativo.

2.5.1 *Cadence Design Framework*

O ambiente de projeto *Cadence Design Framework*¹ é composto por um conjunto de ferramentas que auxiliam o projetista em praticamente todos os procedimentos que compõem o fluxo de projeto de CIs. Esse conjunto de ferramentas consiste em vários programas para diferentes aplicações, tais como desenho de esquemático, simulação, *layout*, verificação, entre vários outros. Na *Cadence*, a integração de todas estas ferramentas é feita por uma ferramenta chamada *Design Framework* (DFW) (Molund, Stefan, 2017). A aplicação DFW é portanto, podemos afirmar, o próprio ambiente *Cadence* pois ela fornece uma interface de usuário e uma base de dados comum às ferramentas utilizadas. Isto faz com que seja possível alternar entre diferentes aplicações sem a necessidade de converter a base de dados.

O ambiente *Cadence* é executado em terminais *Unix* ou computadores com sistema operacional *Linux* instalado (ou emuladores de terminal *Unix*). É necessário ter uma conta criada no computador onde a ferramenta está instalada. Alternativamente, um emulador *Unix* (*Putty*², por exemplo), poderá ser utilizado para acessar remotamente o computador através do protocolo SSH³. Opção ainda, caso o emulador não suporte conexões SSH, é iniciar uma sessão remota (terminal de serviço) utilizando o NXFree⁴.

2.5.1.1 *Desenvolvimento dos Projetos*

Na *Cadence*, conforme ilustra a Figura 2.4, a interface do usuário é gráfica e baseada em janelas e menus com o intuito de facilitar a sua utilização. O projeto de circuitos têm uma estrutura definida, hierárquica, consistindo em bibliotecas, visões e instâncias.

De outra forma, todos os dados de um projeto na *Cadence* são organizados em bibliotecas. Existem bibliotecas de referência que contém blocos básicos de construção e são reutilizáveis na construção de novas bibliotecas em outros projetos.

Cada biblioteca é composta de células e diferentes visões, como mostrado na Figura 2.5. Uma célula é um objeto de banco de dados que forma um bloco de construção, por exemplo, um inversor. Uma visão representa algum nível de abstração da célula, podendo também ser apenas um símbolo representativo, um desenho esquemático do circuito, ou ainda o *layout* que mostra uma visão "tridimensional" do circuito (ou melhor, sobreposição de máscaras para o processo de fabricação) (Molund, Stefan, 2017).

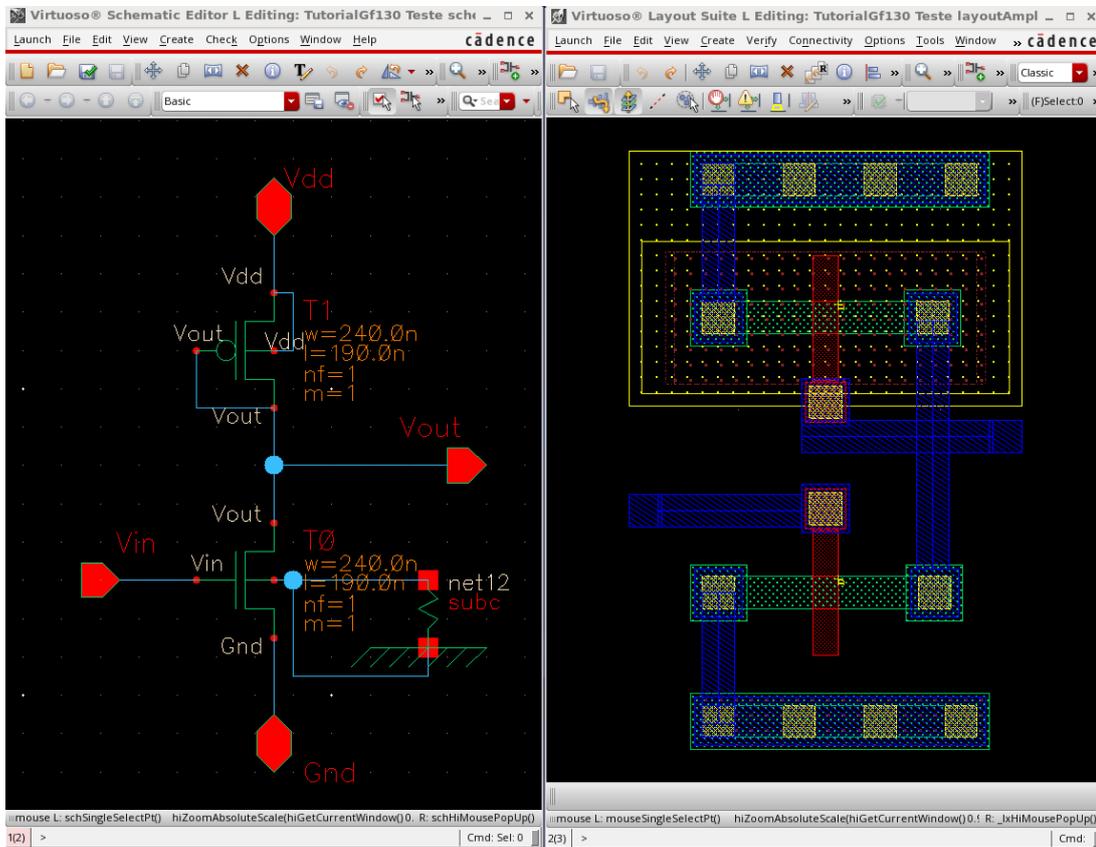
¹ou *Cadence* apenas

²<<http://www.putty.org/>>

³<<https://www.ssh.com/ssh/>>

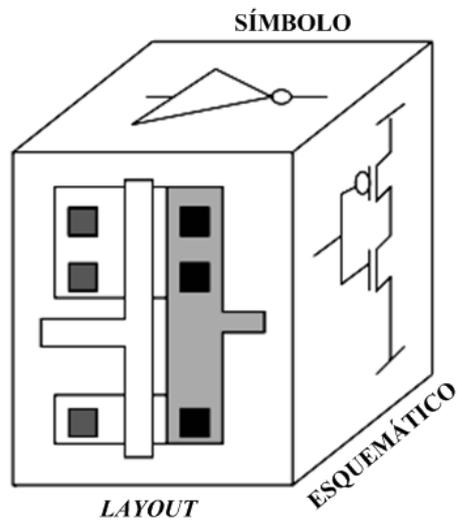
⁴<<https://www.nomachine.com/>>

Figura 2.4 – Ambiente da Cadence: Virtuoso (a) Schematic and (b) Layout Editor.



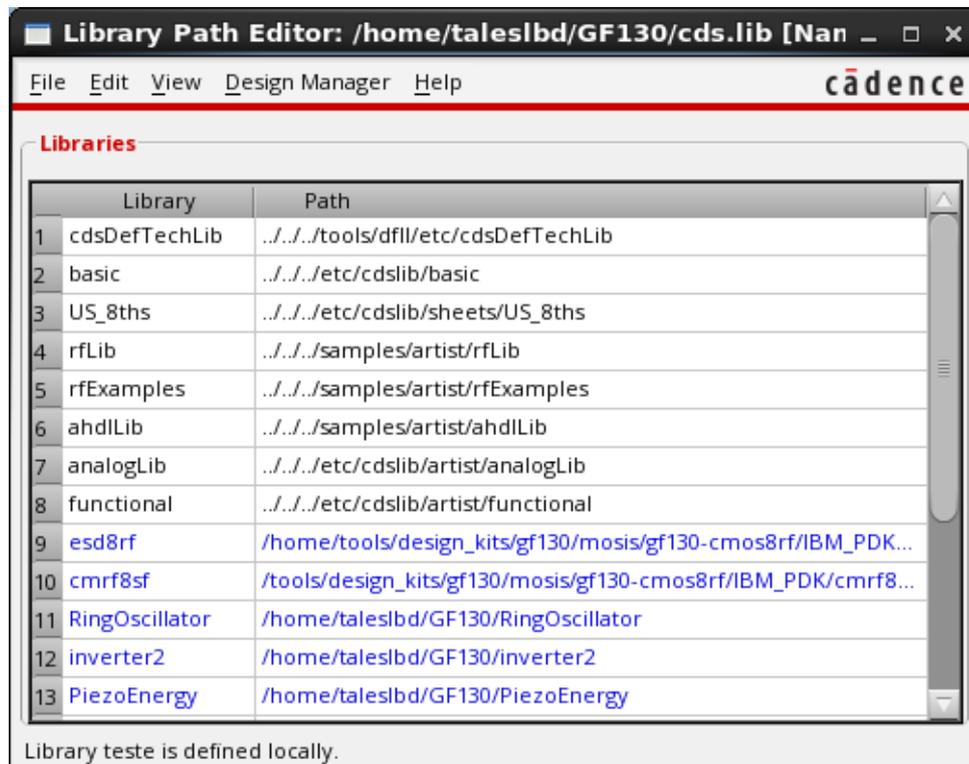
Fonte: Próprio Autor.

Figura 2.5 – A célula de um inversor com três visões: esquemático, símbolo e layout.



Fonte: Adaptado de Molund, Stefan (2017)

Figura 2.6 – Células e bibliotecas instanciadas figuram no arquivo "cds.lib", juntamente como outras bibliotecas.



Fonte: Próprio Autor.

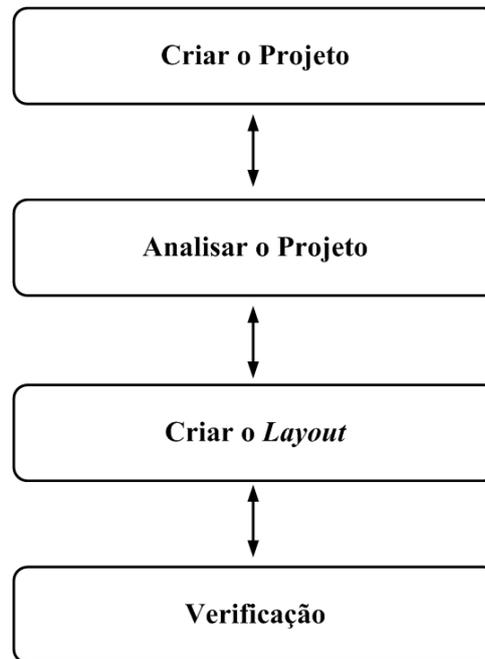
2.5.1.2 Instâncias e Hierarquias

As principais razões para a utilização de modelos hierárquicos é economizar tempo de desenho e minimizar o tamanho da base de dados. Considere um projeto que precise de 500 portas do mesmo tipo. Então, em vez de construí-la 500 vezes, ela é projetada uma única vez e então reutilizada quando for necessário. Desta forma, uma célula pode ser utilizada várias vezes (sem a necessidade de copiá-la) pois a cada utilização a mesma é reinstanciada. A célula a ser instanciada também pode ser de outro projeto ou até mesmo de outro usuário, porém um caminho (endereço do diretório onde está armazenada) precisa estar definido. Esse caminho fica armazenado dentro de um arquivo padrão com o nome de "cds.lib", conforme ilustra a Figura 2.6. Depois de instanciada, apenas a visão do símbolo é apresentada. Essa é uma das principais funcionalidades da *Cadence*, porém necessita de uma correta organização e gerenciamento das bibliotecas para que o compartilhamento se torne eficaz durante projetos em equipe (Molund, Stefan, 2017).

2.5.1.3 Fluxo de Projeto de CIs

O fluxo de projeto simplificado, mostrado na Figura 2.7, mostra as etapas fundamentais do projeto de CIs, desde a concepção inicial até a sua verificação. O fluxo é assumido para o ambiente *Cadence*, ainda que o mesmo seja similar em outros ambientes.

Figura 2.7 – Fluxo de projeto simplificado de CIs..



Fonte: Adaptado de Molund, Stefan (2017)

A etapa Criar o Projeto, que sucede a concepção do CIs, consiste em criar o esquemático de todas as células (feito no Editor de Esquemáticos *Virtuoso Schematic Editor*) e blocos. A visão do esquemático contém símbolos como transistores e talvez outros componentes, como resistores, capacitâncias e linhas de conexão. A partir da visão do esquemático é criada (pode-se gerar quase automaticamente) a visão do símbolo, de modo que a célula possa ser utilizada num nível mais elevado na hierarquia.

A etapa Analisar o Projeto inclui a verificação funcional (ou simulação elétrica) do projeto em nível do esquemático. Simuladores estão integrados ao ambiente para proporcionar fluidez na verificação.

A terceira etapa, Criar o *Layout*, é feito no *Virtuoso Layout Editor*, uma das ferramentas contidas na *Cadence*. Aqui as camadas semicondutoras finais são representadas com cores diferentes. Todas as células e os blocos utilizados têm o tamanho que terá no chip final.

O último passo é a Verificação do projeto, onde o *layout* é examinado sobre infrações contra as regras geométricas ou elétricas, e para verificar a função da implementação física.

2.6 TRABALHOS RELACIONADOS

Para o desenvolvimento desta pesquisa, alguns trabalhos foram analisados a fim de utilizar o conhecimento adquirido a partir da experiência desses estudos para a formulação desta proposta. Os trabalhos relacionados abordam tanto o ensino baseado em equipe em

disciplinas práticas em cursos de engenharia elétrica, eletrônica e computação, quanto o uso de ambientes virtuais colaborativos para o apoio a projetos.

Por fim, para apresentar a relevância e contextualizar esta dissertação, apresentamos a contribuição científica deste trabalho em comparação aos demais.

2.6.1 *Circuits and Systems Engineering Education Through Interdisciplinary Team-based Design Projects*

Dehaene et al. (2011) apresenta um estudo sobre o ensino de circuitos e sistemas através de projetos interdisciplinares baseado em equipe realizado com alunos do curso de engenharia elétrica Universidade Católica de Leuven (Bélgica). O estudo consistiu em um projeto acadêmico realizado durante um ano por um grupo de aproximadamente 20 alunos. Esses alunos foram ensinados a trabalhar como uma equipe multidisciplinar com base em subequipes que lidam com os diferentes problemas de projeto de forma multidisciplinar. Assistentes de ensino foram adicionados à equipe para servir como projetistas especialistas ou como líderes de projeto de uma equipe. Desta forma, os alunos tinham a quem recorrer em caso de dúvidas durante o desenvolvimento dos circuitos e bibliotecas, mas também aprenderam como trabalhar em equipe.

Ao final, os autores apresentam vantagens e desvantagens referentes ao estudo realizado. Como vantagens, o estudo mostrou que os alunos sentiram-se motivados durante a realização do projeto. Ao mesmo tempo, o aprimoramento dos conhecimentos e habilidades técnicas se deu durante as atividades práticas das projeções e simulações dos circuitos e sistemas. Os autores ainda observam que em dez anos de experiência utilizando este método, o resultado e retorno dado pelos alunos são muito positivos. Como desvantagem o autor cita que o método de ensino baseado em equipe exige muito mais horas-pessoa do que o método tradicional teórico-prático. Outra desvantagem é referente à dificuldade de se estimar a carga horária exata para aplicação deste método, pois depende muito da qualidade dos lados que compõem a equipe e dos problemas enfrentados durante a realização do projeto.

2.6.2 *A Teamwork-based Education Strategy for Teaching Lab of Analog Integrated Circuits Design*

Neste trabalho, Rosa (2012) descreve as atividades de educação realizadas para motivar estudantes matriculados em um curso de laboratório focado no projeto de circuitos integrados analógicos. Este estudo foi aplicado em uma disciplina chamada Laboratório de Circuitos Microeletrônicos Avançados, que faz parte do curso de graduação em Engenharia Elétrica e Eletrônica da Universidade de Sevilha (Espanha). Em vez de usar o método tradicional com base em exercícios práticos, o autor seguiu uma metodologia de trabalho em equipe que consistiu no projeto de um CI por um grupo de estudantes, cada

um desempenhando um papel diferente na equipe.

O estudo foi realizado durante um período de quinze semanas, sendo três horas/aula por semana e teve como objetivo propor uma metodologia de ensino que permitisse o estudante aprender através de um trabalho em equipe prático, tentando simular um ambiente de trabalho real realizado em indústrias e fábricas.

Uma avaliação preliminar foi realizada no início do curso a fim de atribuir papéis e tarefas para cada estudante baseado no seu conhecimento. Um dos estudantes foi selecionado como gerente do projeto, enquanto os professores atuaram como revisores do projeto. Foi estabelecido um conjunto de mecanismos de controle como videoconferências e bate-papos, através da plataforma e-learning WebCT ("*Web Course Tools*"), utilizada como ferramenta de apoio a metodologia proposta. Essa ferramenta permitiu que os alunos coordenassem suas atividades, mas também foi utilizada por professores para verificar o trabalho em andamento durante o curso, além das sessões de revisão que se realizavam na sala de aula.

Como resultado dessa combinação de metodologia de trabalho em equipe e aprendizagem colaborativa baseada em projeto, os estudantes se tornaram mais encorajados a realizar os exercícios práticos necessários para cumprir os objetivos do curso e, ao mesmo tempo, desenvolveram e melhoraram outras competências complementares, como a capacidade de trabalhar em equipe e defender seu trabalho por meio de apresentações orais. Rosa (2012) ainda conclui que os alunos que realizaram essa disciplina de laboratório ficaram amplamente satisfeitos e motivados com a experiência de aprendizagem proposta e obtiveram boas notas.

2.6.3 Projeto Cooperativo no Ambiente CAVE

Em Sawicki, Indrusiak e Reis (2002) um ambiente de integração de ferramentas para o projeto de circuitos integrados, denominado CAVE, foi proposto. O objetivo é reduzir o tempo perdido pelo projetista com a administração de recursos através de uma interface gráfica padronizada. O ambiente foi desenvolvido com base no *World Wide Web* (WWW), que além de facilitar a utilização de suas interfaces de controle de rede, possibilita o projeto cooperativo no caso de uma equipe distribuída de projetistas. O ambiente CAVE reúne vários componentes, desenvolvidos em Java, cada um deles associado a uma ou mais etapas de projeto do circuito.

Sua interface simples e padronizada permite uma fácil adaptação do projetista ao ambiente, uma vez que as interfaces das ferramentas de CAD são diferentes umas das outras. Por ser *web*, o ambiente é considerado multiplataforma, pois pode ser acessado de qualquer computador com requisitos mínimos de *software* e *hardware*, necessitando apenas de um navegador *web*. O trabalho remoto via *web* permite o trabalho colaborativo de equipes de projeto dispersas geograficamente. O ambiente CAVE também possui

integração com componentes de *Groupware* como compartilhamento de documentos e comunicação através de *chat*.

Os autores ainda destacam algumas vantagens encontradas durante os projetos desenvolvidos em equipe no ambiente CAVE como a tomada de decisões em equipe, maior harmonia entre os participantes da equipe durante as atividades realizadas, reduzindo o número de erros, trabalhar no mesmo projeto simultaneamente e a fácil adaptação através da interface padronizada, não havendo a necessidade de o usuário perder tempo com as interfaces das ferramentas de CAD.

2.6.4 Ferramenta para gerenciamento e apoio ao desenvolvimento distribuído de software

Wolf e Silva (2013) apresentam o sistema CollabCode, uma aplicação *web* que possui um conjunto de ferramentas para comunicação, gerenciamento de atividades e colaboração para equipes de desenvolvimento distribuídas geograficamente. O CollabCode fornece em uma solução todas as ferramentas necessárias para que equipes geograficamente distribuídas possam trabalhar, sem a necessidade de instalação de qualquer aplicação na estação de trabalho, requerendo apenas um navegador e acesso à Internet.

O sistema foi projetado considerando o Modelo 3C de Colaboração. Isso faz com que o CollabCode seja uma solução que se preocupa diretamente com a comunicação e a cooperação entre os membros da equipe, além de permitir o gerenciamento de atividades inerentes ao projeto. A ferramenta é uma aplicação *web* desenvolvida em PHP (*Hypertext Preprocessor*) com suporte ao banco de dados PostgreSQL. Toda a estrutura do sistema CollabCode foi criada utilizando o Sistema Operacional *Linux* e tem integração com o gerenciador de versões *SubVersion*. Dentre os principais objetivos do sistema desenvolvido, destacam-se dois. O primeiro é fornecer o ferramental necessário para que o desenvolvedor consiga codificar dentro do navegador sem a necessidade de utilização de qualquer outra ferramenta. Para isso, a solução dispõe de uma interface de desenvolvimento com a possibilidade de trabalhar com versionamento de arquivos e utilizar interface com banco de dados. O segundo objetivo é criar uma área de gerenciamento que possibilite o desenvolvimento de projetos *web*, que utilizam como base a linguagem de programação PHP, e o gerenciamento das atividades inerentes ao projeto.

As ferramentas de comunicação existentes no CollabCode permitem que a comunicação entre os membros da equipe ocorra de diversas formas. As ferramentas de comunicação síncronas, como chat e videoconferência, servem de apoio para que todos os envolvidos possam trocar informações de forma online e no momento em que for necessário. Já as ferramentas de comunicação assíncronas, como comentários em atividades, ajudam a identificar as interações dos usuários com as atividades e fornecem subsídios para a documentação do sistema que sendo desenvolvido.

O gerenciamento e a coordenação são possíveis através das ferramentas de gerenciamento de projetos e gerenciamento de atividades. O gerenciamento de projetos dispõe de uma interface para criação e gerenciamento de projetos. Ao adicionar um projeto, o usuário pode optar por importar um projeto por meio de repositório *SubVersion*⁵ ou criar um novo projeto. O gerenciamento de atividades dispõe de uma interface em que é possível criar e gerenciar atividades, além de permitir seu acompanhamento por meio de gráficos de *Gantt*. Essa ferramenta possibilita também o acompanhamento da execução das atividades por meio de um percentual de conclusão e dos comentários adicionados pelos membros da equipe.

A colaboração entre os membros é possível através das ferramentas edição colaborativa, compartilhamento de tela e quadro branco. A edição colaborativa de código pode ser compartilhada entre os desenvolvedores. Caso dois ou mais desenvolvedores abram o mesmo arquivo, uma sessão de edição colaborativa é iniciada, permitindo-lhes a edição cooperativa em tempo real do código fonte. O compartilhamento de tela permite que um membro da equipe possa compartilhar área de trabalho com os demais membros da equipe. A ferramenta permite também que o computador que está sendo compartilhado possa ser controlado remotamente por outro membro da equipe. O quadro branco é um recurso disponível na ferramenta de videoconferência. Com ele é possível que os membros da equipe interajam em tempo real na construção de artefatos de *software* que facilitem a compreensão de processos e de atividades a serem desenvolvidas.

Os autores concluem que existem diversas ferramentas para o desenvolvimento colaborativo, porém nenhuma oferece todos os recursos necessários para tal. Nesse sentido, CollabCode é uma opção às atuais ferramentas existentes no mercado. Sua estrutura foi projetada de modo a tornar o processo de desenvolvimento de software mais ágil e facilitar a coordenação, colaboração e comunicação em equipes.

2.6.5 Contribuição deste trabalho

Face às necessidades do trabalho colaborativo, do projeto em equipe de CIs e de suas particularidades, relevando os trabalhos relacionados na literatura, estabeleceu-se como contribuição desse trabalho a concepção, projeto e implementação de um ambiente virtual colaborativo para apoiar o ensino de CIs baseado em equipe. O ambiente deve fornecer ao projetista meios mínimos para lidar com as necessidades e desafios encontrados durante os projetos em equipe com foco acadêmico, o que, conforme demonstram os trabalhos de Dehaene et al. (2011) e Rosa (2012), melhora o aprendizado dos participantes, ensinando-os não apenas as atividades práticas inerentes ao conteúdo, mas também competências complementares como a capacidade de trabalhar em equipe e liderança.

O ambiente virtual colaborativo proposto neste trabalho, além de servir como um

⁵<https://subversion.apache.org/>

ambiente de apoio ao ensino baseado em equipe como visto em Dehaene et al. (2011) e Rosa (2012), fornece ferramentas de colaboração que, assim como as propostas de Sawicki, Indrusiak e Reis (2002) e Wolf e Silva (2013), garantem a organização dos projetos e uma cooperação eficiente entre os participantes, aprimorando práticas individuais, a troca de conhecimento entre a equipe e também o papel da liderança.

Apesar da similaridade com os estudos mencionados (SAWICKI; INDRUSIAK; REIS, 2002) (WOLF; SILVA, 2013), este trabalho se diferencia pela sua simplicidade (e eficiência decorrente) e facilidade durante a sua execução. Sua estrutura hierárquica permite uma melhor organização dos arquivos que compõem as bibliotecas desenvolvidas dentro de um projeto. O controlador de versões, baseado em um modelo distribuído, além de permitir aos participantes o trabalho e controle de suas atividades individualmente, também possibilita realizá-los simultaneamente com os demais participantes do mesmo projeto. O controlador de versões possui funcionalidades essenciais como a submissão e recuperação de versões através de uma interface *web* descomplicada, não necessitando de programa extra ou de digitação auxiliar de comandos, o que poderia resultar em dificuldades ao projetista para se situar no contexto colaborativo. Ferramentas de comunicação permitem a troca de ideias e de conhecimento durante a realização das atividades. O compartilhamento de materiais de apoio e sugestões de projeto pode ser realizado a qualquer momento. Por fim, sempre que necessário, o ambiente permite ao coordenador a compilação de um determinado projeto, realizando a cópia de todas as bibliotecas para um determinado repositório onde poderá ser compartilhado para todos os usuários do ambiente e utilizado em outro projeto.

3 IC-COLLAB: AMBIENTE VIRTUAL COLABORATIVO PARA APOIO AO ENSINO DE PROJETOS DE CIRCUITOS INTEGRADOS EM EQUIPE

Neste capítulo será apresentado o processo de análise, modelagem e construção do Ambiente Virtual Colaborativo para Apoio ao Ensino de Projeto de Circuitos Integrados em Equipe, denominado IC-COLLAB.

Durante o desenvolvimento do ambiente, visando diminuir a complexidade e garantir um *software* de qualidade, foi utilizado o Ciclo de Desenvolvimento de *Groupware*. Em seguida, é apresentada uma visão geral do ambiente, destacando sua organização e ferramentas disponíveis. Por fim, todas as funcionalidades do IC-COLLAB são detalhadas.

3.1 MODELAGEM E PROJETO DO AMBIENTE VIRTUAL IC-COLLAB COM BASE NO CICLO DE DESENVOLVIMENTO DE *GROUPWARE*

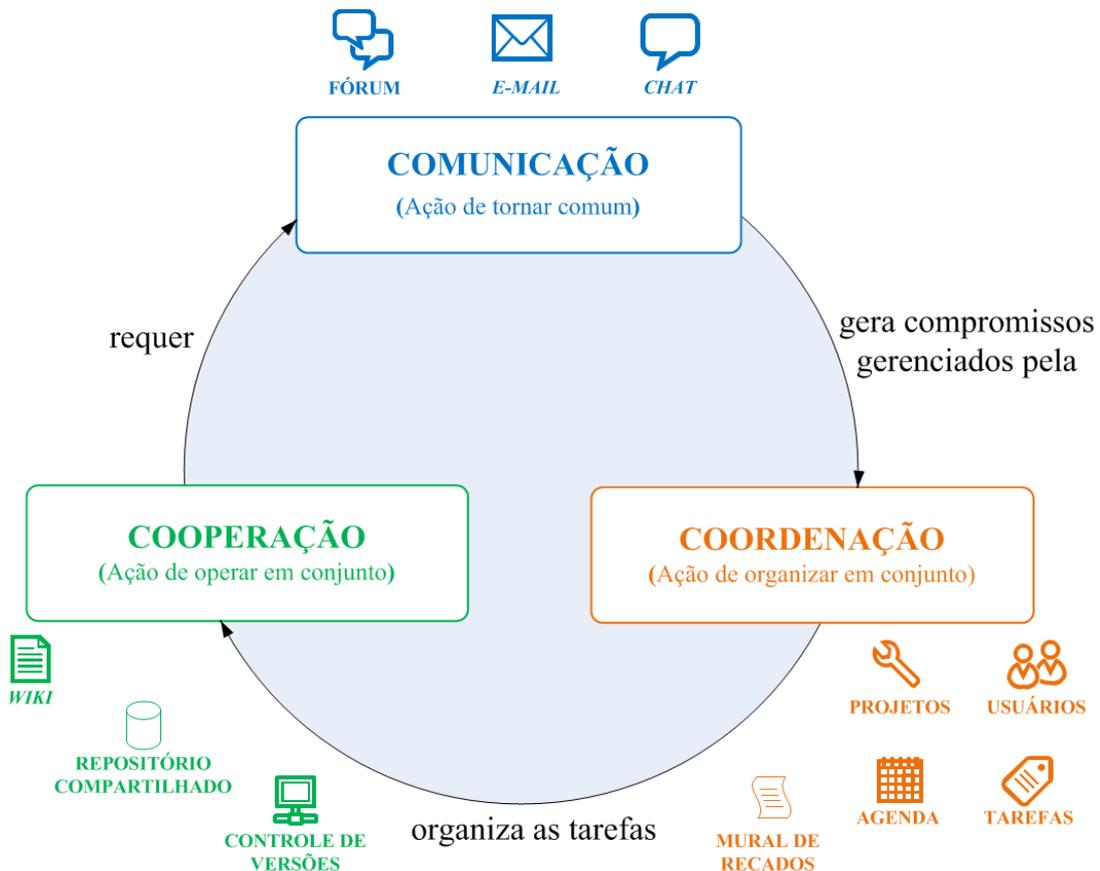
Geralmente, o desenvolvimento de um novo ambiente virtual inicia porque os existentes já não mais atendem às necessidades de um determinado grupo (BOTELHO; VIDAL, 2005), ou ainda, simplesmente porque ainda não existe tal ambiente. No presente trabalho, de fato não identificou-se na literatura ambiente similar ao que se desejava, com as funcionalidades que se buscava.

A ideia do IC-COLLAB é fornecer em uma única solução todos os componentes necessários para que equipes possam colaborar com esforços mínimos, seja presencial ou distribuído geograficamente e sem a necessidade de instalação de qualquer aplicação na estação de trabalho, requerendo apenas um navegador e acesso a *internet*. Desta forma, optou-se por desenvolver uma aplicação *web* com um conjunto de componentes que permitam a comunicação, coordenação e cooperação. Com base na literatura (BRITO; PEREIRA, 2004)(PIMENTEL et al., 2006)(SAWICKI; INDRUSIAK; REIS, 2002)(WOLF; SILVA, 2013)(GILAB, 2017)(Subversion, 2018), os componentes foram definidos como segue:

- **Comunicação:** *chat*, *e-mail* e fórum;
- **Coordenação:** gerência de usuários, tarefas, agenda, mural de recados e gerência de projetos;
- **Cooperação:** repositório compartilhado, controle de versões e *wiki*.

Com o propósito então de se desenvolver o ambiente virtual IC-COLLAB, optou-se pelo Ciclo de Desenvolvimento de *Groupware* proposto por Fuks et al. (2003), que através do seu processo sistemático dividido em fases, mostrou-se o mais adequado para o processo de construção do ambiente em questão. A seguir cada uma das fases será detalhada.

Figura 3.1 – Componentes do IC-COLLAB classificados conforme Modelo 3C.



Fonte: Próprio Autor.

3.1.1 Análise de Domínio

Nesta primeira fase, o analista e/ou o programador preocupou-se em analisar o ambiente onde o *software* será aplicado, pré-definindo e classificando informações relevantes com base em estudos realizados juntamente com os usuários (alunos e professores) dos laboratórios do GMICRO e estudos de outros ambientes virtuais (PRESMAN, 2011). Para tal, o Modelo 3C serviu de instrumento para o desenvolvedor durante esse processo de análise e deste modo, cada componente do IC-COLLAB foi classificado em função do grau de suporte à comunicação, coordenação ou cooperação, e também a inter-relação entre eles.

A Figura 3.1 ilustra o sistema de classificação adotado para a organização dos componentes de colaboração do IC-COLLAB. Oportuno esclarecer que os componentes estabelecidos para compor o ambiente IC-COLLAB advém das necessidades fundamentais de colaboração identificados durante as etapas de projetos de CIs no ambiente Cadence. Estão a seguir descritos juntamente com a referida análise de domínio:

- *Chat*

Classificado como componente da dimensão **Comunicação** (síncrona), o *Chat* pos-

sibilita visualizar quais os participantes estão *online* e também a troca de mensagens instantâneas entre eles. Através da troca de mensagens é possível coordenar (**Coordenação**) as ações da equipe, delegando tarefas e acompanhando o andamento das mesmas. A troca de mensagens também é um meio de compartilhar informações (**Cooperação**) durante o trabalho colaborativo, dividindo novas ideias tomando decisões em equipe. Do mesmo modo, esse compartilhamento de informações através do *chat*, são registradas no ambiente virtual para consultas futuras (IONESCU et al., 2015).

- *E-mail*

O *E-mail* (ou correio eletrônico) é um componente que permite enviar e/ou receber mensagens de texto entre os participantes de uma equipe de trabalho, e é portanto classificado com a dimensão **Comunicação** (assíncrona). Entretanto, ele pode também prover **Coordenação** através de mensagens de texto, com instruções sobre determinada tarefa ou atividade, enviadas para um ou mais participantes da equipe. Da mesma forma, a **Cooperação** é também provida quando tarefas e atividades são concluídas e suas soluções registradas na base de dados do ambiente virtual.

- Fórum

O componente Fórum é classificado com a dimensão **Comunicação** (assíncrona), porém através dele é também possível coordenar (**Coordenação**) a equipe através de mensagens de texto publicadas em listas de discussão sobre assuntos de grande importância a todos os participantes (QI et al., 2010). Durante o debate ocorre o compartilhamento de informações (**Cooperação**) através da discussão sobre o tema do problema em questão, registrando tudo na base de dados do ambiente virtual.

- Usuários

Este componente permite a gerência de usuários dentro do ambiente IC-COLLAB. Classificado com a dimensão **Coordenação**, é durante o registro de usuários que serão introduzidas as informações necessárias para a **Comunicação** entre elas (como por exemplo, o endereço de *e-mail*). Importante ressaltar que é também durante esse registro que o repositório local de trabalho do usuário é definido. Ainda, é através desse componente que torna-se possível compartilhar (**Cooperação**) arquivos referentes a bibliotecas desenvolvidas com os demais participantes do projeto.

- Repositório compartilhado

Esse componente provê um espaço compartilhado que permite aos participantes de um mesmo projeto cooperarem (**Cooperação**) (QI et al., 2010). Da mesma maneira, o repositório pode ser gerenciado (**Coordenação**), pois o participante com função de coordenador é responsável por distribuir as tarefas inerentes a cada biblioteca

dos demais participantes. A troca de informações é realizada a qualquer momento utilizando os componentes *Chat* e/ou *E-mail* (**Comunicação**).

- Controle de versões

O Controle de versões é um componente classificado com a dimensão **Cooperação** por ser capaz de controlar e compartilhar as mudanças realizadas nas bibliotecas. A cada mudança uma versão é criada e fica armazenada no repositório local do participante e, caso necessário, disponível a todos os outros. A gerência das versões possibilita ao coordenador (**Coordenação**) analisar arquivos de uma biblioteca em busca de inconsistências ou mesmo erros, e nesses casos, delegar tarefas ao participante para que este encaminhe solução.

- Tarefas

O componente Tarefas é classificado com a dimensão **Coordenação**. Ele permite o registro e gerência de tarefas pelo coordenador do projeto. A troca de informações **Comunicação** é realizada no momento em que o coordenador registra uma determinada tarefa para um participante do projeto, fazendo com que um *e-mail* seja disparado pelo ambiente virtual para informá-lo sobre a atividade. À medida que o participante registra o andamento das atividades, atualizando ou finalizando a tarefa delegada, o *e-mail* é disparado informando ao coordenador e demais participantes sobre a situação. A **Cooperação** é possível através do componente *Chat*, *E-mail* ou pelo Fórum, onde os participantes podem trocar mensagens e discutir soluções para determinadas tarefas.

- Agenda

O componente Agenda possibilita a organização de Tarefas e informações sobre as atividades realizadas dentro do ambiente virtual. Classificada com a dimensão **Coordenação**, o componente permite informar, através de data e hora, sobre o prazo para se concluir determinada tarefa e também sobre as atividades realizadas pelos participantes. Por ser compartilhada (Cooperação) no ambiente virtual, a Agenda torna o trabalho da equipe transparente, disponibilizando a todos (**Comunicação**) informações das tarefas de cada um e prazos de execução associados.

- Mural de recados

Componente classificado com a dimensão **Coordenação**, o Mural de recados permite que o coordenador registre recados para a equipe (QI et al., 2010). Ao registrar um recado, um *e-mail* (**Comunicação**) é disparado para toda a equipe. Da mesma maneira, o recado fica compartilhado (**Cooperação**) na tela inicial de cada participante.

- *Wiki*

Componente que permite o compartilhamento (**Cooperação**) de textos explicativos ou até mesmo arquivos contendo informações de ajuda sobre determinado assunto ou problema encontrado em um projeto (SONEGO et al., 2014). Assim como no componente Mural de recados, quando um novo texto é registrado, um *e-mail* é enviado (**Comunicação**) a toda a equipe. Os textos podem ser editados por qualquer participante do projeto. Caso o coordenador encontre um texto ou arquivo que não esteja de acordo com o projeto, o mesmo é editado ou removido (**Coordenação**).

- Projetos (Gerência de Projetos)

Sendo um dos principais componentes no IC-COLLAB, a gerência de Projetos é o componente que integra todos os componentes e permite armazenar e gerenciar (**Coordenação**) todas as informações referente ao projeto. Desta forma, todos os componentes são centralizados e disponibilizados em apenas um local no IC-COLLAB, tornando o trabalho transparente, organizado e eficiente (**Cooperação e Comunicação**).

3.1.2 Análise de Requisitos

Na fase de Análise de Requisitos, procedeu-se a descrição dos requisitos do sistema e dos seus serviços, levando-se em conta seus objetivos (e limitações) e os componentes definidos para o ambiente. O processo de análise e levantamento dos requisitos foi realizado juntamente com os usuários (alunos e professores) dos laboratórios do GMICRO. Inicialmente, desenvolveu-se a compreensão do domínio onde o IC-COLLAB será aplicado e em seguida a coleta dos requisitos foi realizada através de entrevistas com os usuários do GMICRO. Por fim, todos os requisitos foram revisados a fim de verificar se estão em concordância com o que foi coletado.

Apesar de não ser possível especificar completamente o ambiente IC-COLLAB através da lista de requisitos, ela auxiliou e foi fundamental para o processo de desenvolvimento.

Entre os principais requisitos levantados, pode-se mencionar os processos de cadastro usuários no sistema, cadastro e gerenciamento dos projetos, submeter alterações das bibliotecas, entre outros.

Listagens mais completas dos requisitos funcionais e dos não-funcionais do Sistema IC-COLLAB são apresentadas no Anexo A e B, respectivamente.

3.1.3 Projeto

Na fase de Projeto, os processos do ambiente IC-COLLAB foram modelados e documentados com base nos requisitos levantados na fase anterior. A modelagem e documen-

tação de um *software* auxiliam o desenvolvedor a compreender melhor os seus processos através de diagramas (SOMMERVILLE, 2011).

Para a documentação, optou-se pela linguagem UML (*Unified Modeling Language*) por possibilitar a inclusão de definições através de diagramas de modelagem para as diversas etapas do desenvolvimento, desde as iniciais (mais abstratas) até as finais e mais refinadas. A arquitetura geral e tecnologias utilizadas também são apresentadas nas subseções a seguir.

3.1.3.1 Diagrama de Caso de Uso

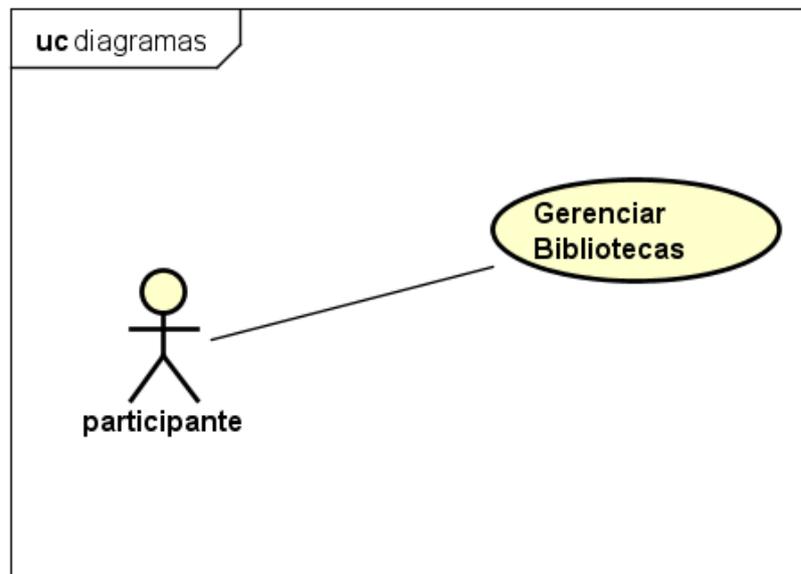
O Diagrama de Caso de Uso tem o objetivo de auxiliar a comunicação entre os analistas e/ou desenvolvedores e o cliente, descrevendo através de diagramas os processos de um novo sistema que será projetado. Este diagrama ilustra graficamente uma visão externa do sistema através de três entidades: atores, casos de uso e o relacionamento entre elas (SOMMERVILLE, 2011). O ator especifica um papel executado por um usuário que interage com o sistema. O caso de uso é um conjunto de ações executadas pelo sistema e o relacionamento representa a interação e associação entre eles. Conforme ilustrado na Figura 3.2, o diagrama de caso de uso é exemplificado através de um processo modelado para o IC-COLLAB. O "participante" é o ator representado pelo boneco que se relaciona com o caso de uso "Gerenciar Bibliotecas", representado pela elipse. O diagrama contendo todos os casos de uso modelados para o ambiente IC-COLLAB, com base na lista de requisitos, é apresentado no Anexo C.

3.1.3.2 Diagrama de Sequência

O diagrama de sequência representa a sequência de ações que necessitam ser executadas em um sistema seguindo uma ordem lógica no tempo. Tem o objetivo de mostrar como as mensagens entre os objetos são trocadas no decorrer do tempo para a realização de uma ação (SOMMERVILLE, 2011). O diagrama de sequência é composto por atores, objetos e mensagens, sendo os objetos representados por retângulos sobre uma linha tracejada denominada linha da vida, que representa o tempo de vida do objeto na interação do sistema. A mensagem é representada por uma linha com uma seta na ponta, indicando a direção da mensagem entre dois objetos (MALUCELLI et al., 2010).

A Figura 3.3, apresenta um exemplo de diagrama de sequência de processos do IC-COLLAB, no qual o coordenador cadastra uma biblioteca para um determinado participante do projeto, que em seguida começa a realizar alterações. De acordo com o exemplo, o "coordenador" do projeto no papel de ator envia a mensagem "cadastrar biblioteca" para o objeto "formBibliotecas" e em seguida recebe a mensagem de retorno "cadastro OK", que informa que uma nova biblioteca foi cadastrada com sucesso. Na sequência, o sistema atribui a biblioteca para o "participante" (outro ator) enviando a mensagem "cadastra

Figura 3.2 – Diagrama de caso de uso representando graficamente uma funcionalidade do IC-COLLAB.



powered by Astah

Fonte: Próprio Autor.

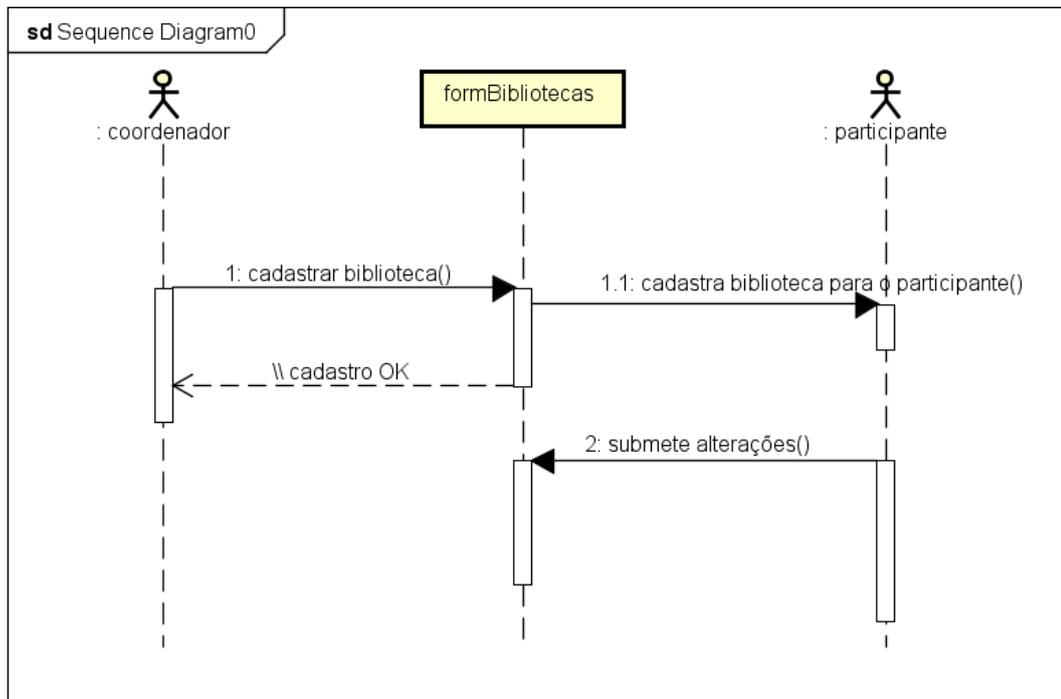
biblioteca" para o "participante", que a partir disso pode submeter o seu trabalho desenvolvido, indicado pela mensagem "submete alterações". Na sequência, o ator participante recebe a mensagem "cadastra biblioteca para o participante" que indica que ele já pode realizar alterações, retornadas a partir da mensagem "submete alterações". O diagrama de sequência com toda a sequência de processos do IC-COLLAB é apresentado no Anexo D.

3.1.3.3 Diagrama Entidade-Relacionamento

A técnica de modelagem de dados mais utilizada é a abordagem entidade-relacionamento. Nessa técnica, o modelo de dados é representado graficamente através de um diagrama entidade-relacionamento. Uma entidade representa um conjunto de objetos sobre os quais deseja-se manter informações no banco de dados. Uma entidade, por exemplo, seria o conjunto dos usuários de um sistema (COUGO, 2013). No diagrama entidade-relacionamento, a entidade é representada por um retângulo. O relacionamento é uma propriedade que especifica as associações entre ocorrências de entidades.

Um exemplo de relacionamento está mostrado na Figura 3.4, onde se deseja saber quais os projetos (entidade projetos) que são de determinada categoria (entidade categoria). Um relacionamento é representado por um losango, ligado por linhas aos retângulos representativos das entidades que participam do relacionamento (HEUSER, 2009). A partir do diagrama entidade-relacionamento pode-se obter um melhor entendimento sobre o

Figura 3.3 – Diagrama de sequencia representando parcialmente uma sequência de processos do IC-COLLAB.



powered by Astah

Fonte: Próprio Autor.

projeto do banco de dados que será necessário e assim especificar melhor a sua estrutura (SOMMERVILLE, 2011). A estrutura do banco de dados do IC-COLLAB é apresentada no Anexo E através do diagrama entidade-relacionamento.

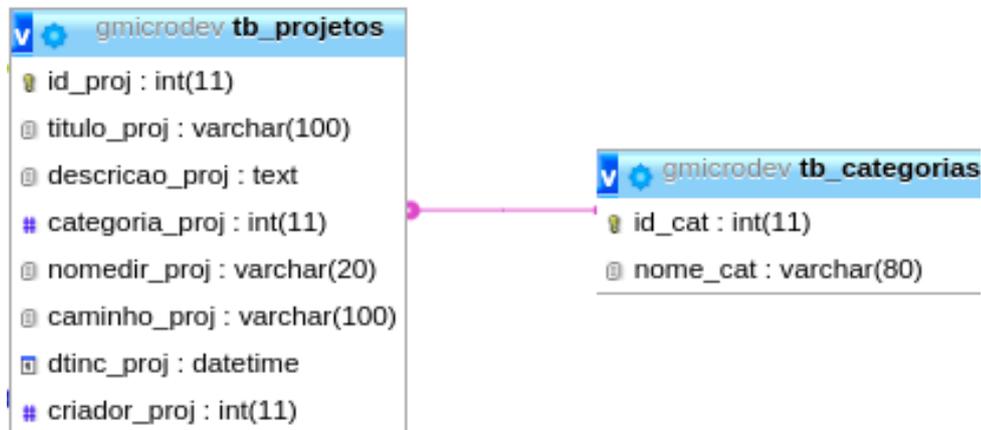
3.1.3.4 Arquitetura

A definição da arquitetura de um sistema, de modo geral, permite a todos os envolvidos (desenvolvedor, analista e futuros usuários) melhor compreender como a infraestrutura do sistema será organizada. (SCHAF, 2011)(HIRAMA, 2011)(JR, 2010) .

A arquitetura é o elo entre as fases Análise de Requisitos e Projeto, pois identifica as principais características estruturais do sistema e os relacionamentos entre elas (PRESMAN, 2011). Alguns padrões de arquitetura (por exemplo, arquitetura em camadas, arquitetura de repositório, arquitetura Cliente-Servidor, etc) são comumente usadas por desenvolvedores e analistas de sistemas para auxiliar no processo de definição da arquitetura de um sistema Sommerville (2011).

Com base nos requisitos levantados anteriormente (por exemplo, Criar base de dados centralizada do projeto), a arquitetura do IC-COLLAB foi definida conforme o padrão Cliente-Servidor. Esta arquitetura é comumente utilizada para o desenvolvimento de sistemas *web*. No padrão Cliente-Servidor, a estrutura é distribuída e a funcionalidade do

Figura 3.4 – Diagrama entidade-relacionamento mostrando como duas entidades se relacionam no IC-COLLAB.



Fonte: Próprio Autor.

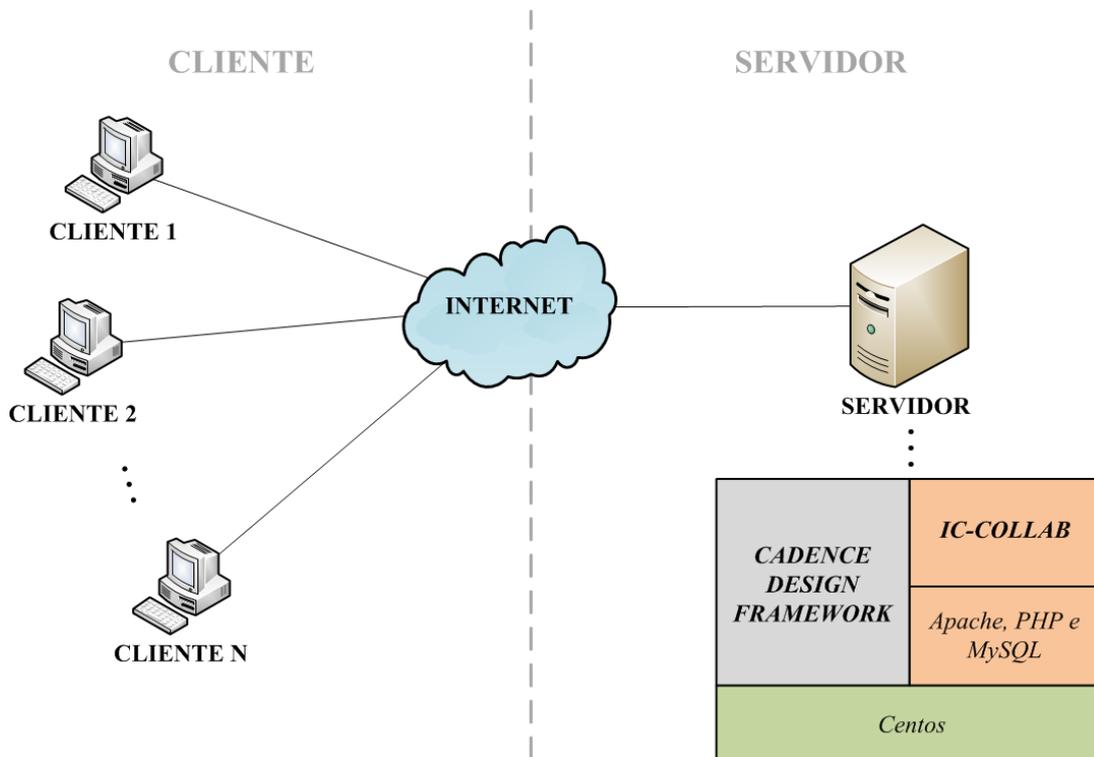
o sistema está organizada em serviços que são prestados a clientes (computadores, *tablets*, celulares) conectados através da rede de computadores e é dividido em duas partes: lado do Servidor e o lado do Cliente. O lado do Servidor possui um ou mais computadores (com alta capacidade de armazenamento e processamento) que oferecem serviços para os clientes, como por exemplo, um serviço de impressão compartilhado ou um serviço de compartilhamento de arquivos. No lado do Cliente, um ou mais clientes podem solicitar os serviços oferecidos pelos servidores. Cada cliente pode enviar requisições de dados para o Servidor e esperar pela resposta. Por sua vez, o Servidor pode aceitar tais requisições, processá-las e retornar o resultado para o Cliente.

Uma visão geral da arquitetura do IC-COLLAB é apresentada na Figura 3.5. No lado do Servidor está o computador (central) onde estão instaladas todas as tecnologias necessárias para o funcionamento do IC-COLLAB. Nesse mesmo computador está instalado o ambiente de projeto de *Cis Cadence*. No lado do Cliente estão os clientes propriamente ditos que realizam as requisições ao servidor através de navegadores *web* instalado em seus computadores, *tablets*, celulares, etc.

3.1.3.5 Tecnologias utilizadas

No presente trabalho, o termo tecnologia refere-se ao conjunto de ferramentas necessárias para o funcionamento do ambiente virtual IC-COLLAB. Por se tratar de um sistema *web*, a definição das tecnologias utilizadas no desenvolvimento do IC-COLLAB está ligada diretamente à necessidade de um repositório centralizado acessível de locais diferentes. É necessário também a garantia de organização e eficiência durante os projetos de CIs. Para tal, as tecnologias detalhadas abaixo foram escolhidas considerando a arquitetura apresentada (Figura 3.5), em se tratando de dados que necessitam estar centralizados para acesso rápido, robusto e seguro.

Figura 3.5 – Arquitetura geral do IC-COLLAB. Cadence e IC-COLLAB estão no lado do Servidor provendo serviços aos clientes (lado dos Clientes) por meio da internet.



Fonte: Próprio Autor.

- Sistema operacional

O sistema operacional utilizado no servidor foi o Linux, na distribuição *Centos*¹ versão 5, visto ser o sistema homologado para executar a ferramenta *Cadence*. O *Centos* é uma distribuição gratuita, com diferencial na estabilidade, robustez, gerenciamento, segurança e confiabilidade (CentOS, 2018).

- Servidor *web*

O servidor *web* é uma tecnologia fundamental para a infraestrutura do IC-COLLAB. Ele é um programa que usa o HTTP (*Hypertext Transfer Protocol*) para servir os arquivos que formam páginas da *web* para os usuários, em resposta aos seus pedidos, que são encaminhadas pelos clientes HTTP de seus computadores. Computadores dedicados e equipamentos podem ser referidos como servidores *web* também. Falando genericamente, o servidor *web* é responsável por armazenar e trocar informações com outras máquinas. Para que isso aconteça ao menos dois participantes estão envolvidos nesta troca de informações: os usuários (clientes) e os servidores (MARCELO, 2005). Diversos servidores *web* são utilizados mundialmente, alguns são pagos e outros são gratuitos, como por exemplo o IIS (*Internet Information Services*) e o Apache.

¹<<https://www.centos.org/about/>>

Para este trabalho, por ser de fácil uso e por atender aos requisitos da arquitetura proposta, o servidor Apache foi escolhido como servidor *web*. Não houve necessidade de configurações adicionais à configuração padrão do Apache (Apache, 2017).

- Linguagem de programação

A linguagem de programação utilizada foi PHP, uma linguagem de *script* pensada para o desenvolvimento de aplicativos *web* dinâmicos, pois ele é executado no lado do servidor e pode ser incorporado ao HTML (*HyperText Markup Language*) (PHP, 2017)(MILANI, 2010). A linguagem HTML é uma linguagem de marcação utilizada na construção e organização de páginas na *web*. Ela é a base da estrutura de uma página da *web*, ou seja, ela é o que o navegador interpreta para apresentar a página ao usuário em frente ao computador (HTML, 2017)².

- *Layout*

O *layout* do sistema foi desenvolvido usando o *framework Bootstrap*. O Bootstrap é um *framework front-end* para o desenvolvimento de *websites* responsivos, ou seja, é um *framework* para organização do conteúdo em determinada página baseado no tamanho das telas dos dispositivos (BOOTSTRAP, 2017). Devido às amplas características dos dispositivos móveis atuais, o *bootstrap* é projetado para a adaptação do conteúdo do *website* em telas de qualquer tamanho através de pontos de interrupção baseados na largura e altura mínima dessas telas. O *bootstrap* possui também uma vasta biblioteca de componentes extras com funcionalidades variadas muito úteis e que podem ser integradas facilmente.

- Banco de dados

Para garantir um armazenamento de dados confiável foi adotado o Banco de Dados MySQL. MySQL é um Sistema de Gerenciamento de Banco de Dados (SGBD) que usa a linguagem SQL (*Structured Query Language*) como interface e permite um uso eficiente de armazenamento, procura, ordenação e recuperação de dados. O MySQL controla o acesso aos dados para garantir que múltiplos usuários possam trabalhar concorrentemente e que somente usuários autorizados possam acessá-los (LEITE; BONOMO, 2017)(MILANI, 2010).

3.1.4 Implementação

Na fase de Implementação, conforme visto anteriormente, a prototipação é realizada a partir do modelo incremental de desenvolvimento de *software* e portanto, a codificação deve sempre resultar em um protótipo que implemente tudo o que foi especificado durante as fases anteriores (Análise de Domínio, Análise de Requisitos e Projeto).

²Não é escopo deste trabalho avaliar o nível de segurança dessas tecnologias e portanto mais informações estão disponíveis em <http://php.net/manual/pt_BR/security.php>

No caso do ambiente IC-COLLAB, que se apresentou um sistema com vários componentes, estes foram implementados a partir da especificação de cada componente e posteriormente integrados (interconectados).

Por fim, o protótipo foi liberado para a fase de Testes. Para fins ilustrativos, utilizou-se os seguintes equipamentos e *software* durante a implementação:

- **SERVIDOR**

Configuração: processador *Intel Xeon* (8 núcleos físicos, 2.33 GHz cada), 16 GB de memória RAM e 1 TB de HD. Este servidor foi utilizado como servidor web e de banco de dados pelo seu alto poder de processamento, e por isso foi instalado a aplicação Apache, responsável por prover um ambiente web sob a arquitetura cliente/servidor, também responsável pela hospedagem do IC-COLLAB. Para completar, foi também instalado o SGBD MySQL, responsável pelo armazenamento de dados;

- **NOTEBOOK**

Configuração: processador *Intel Core i3* 2.20 GHz, 4 GB de memória RAM e 220 GB de HD. Este notebook foi utilizado para a implementação (codificação) do IC-COLLAB através do IDE Geany³. Geany é uma IDE simples, de fácil uso e configuração que permite a codificação em várias linguagens, dentre elas, HTML, PHP e *ShellScript*, utilizadas neste trabalho.

3.1.5 Testes

Na fase de Testes, preocupou-se em avaliar se o sistema estava de acordo com o proposto durante a fase de especificação buscando descobrir eventuais problemas em sua operação antes do uso propriamente dito. Desta forma, neste trabalho, os testes no IC-COLLAB foram realizados pelo desenvolvedor e também por alunos e professores do laboratório do GMICRO.

Primeiramente foram realizados testes de defeitos onde diversas simulações usando dados fictícios foram realizadas, como por exemplo, mensagens emitidas após um cadastro, campos com preenchimento obrigatório, envio automático de *e-mail*, etc. Posteriormente foram realizados os testes de validação, simulando situações reais de projetos, como por exemplo, cadastrar um projeto, adicionar participantes ao projeto, criar bibliotecas para os participantes, criar tarefas para os participantes, realizar o controle de versão de uma biblioteca, etc.

A medida que eram encontrados alguns processos incorretos e/ou erros no IC-COLLAB, o ciclo de desenvolvimento de *groupware* era reiniciado, sendo as vezes necessária a revisão de conceitos e dos requisitos levantados.

³<<https://www.geany.org/>>

3.2 VISÃO GERAL DO IC-COLLAB

O ambiente virtual colaborativo IC-COLLAB é um *software* que agrupa diversos recursos e componentes dos ambiente virtuais conhecidos e abrange de forma equilibrada as três dimensões (comunicação, coordenação e cooperação) do Modelo 3C, fundamentais para prover o trabalho colaborativo. Desenvolvido sobre a arquitetura cliente/servidor, o IC-COLLAB não precisa ser instalado nos computadores dos usuários e deste modo, apenas um computador com acesso à internet e com um navegador instalado é necessário.

Para um trabalho colaborativo flexível, eficiente e organizado, o IC-COLLAB foi projetado com um repositório distribuído de armazenamento baseado na ferramenta Git (CHA-CON; STRAUB, 2017). O repositório distribuído é composto por duas partes: repositório e área de trabalho. O repositório fica no ambiente virtual IC-COLLAB, instalado no servidor central e a área de trabalho no computador local do usuário. O repositório do IC-COLLAB é dividido em duas partes: o repositório principal dos projetos (cada projeto tem o seu repositório), e o repositório local de cada usuário que será a cópia da sua área de trabalho.

Optou-se por um repositório distribuído respeitando a forma usual utilizada para o projeto de CIs, onde associa-se uma biblioteca e o seu desenvolvimento a um único usuário (ao invés de se trabalhar no nível de células). De outra forma, haverá um único projetista para cada biblioteca, ou seja, em nenhum momento haverá necessidade de o sistema controlar edições simultâneas de uma mesma biblioteca. O repositório distribuído faz parte do componente Repositório compartilhado do IC-COLLAB, pois provê um espaço de trabalho organizado hierarquicamente a todos os participantes de um mesmo projeto.

Conforme apresentado na Figura 3.6, sua estrutura distribuída possibilita que os usuários 1 a N possuam seus próprios repositórios locais, o que permite que os projetos 1 a M em equipe sejam estruturados e compilados hierarquicamente no repositório principal. Com esta hierarquia, a estrutura do projeto é transparente para todos os usuários e a colaboração entre os membros da equipe mais eficiente. A Figura 3.6 também demonstra que um determinado usuário pode participar em mais de um projeto (ou seja, a arquitetura é multi-projeto e multi-usuário), sem comprometer a organização. Os participantes de um determinado projeto não trabalham diretamente no repositório local e sim em sua área de trabalho, que fica no computador onde o usuário está projetando as bibliotecas de CIs na *Cadence*. A qualquer momento, os participantes podem sincronizar seu trabalho, armazenado em sua área de trabalho local, com o repositório local do IC-COLLAB. Essa sincronização realiza o controle de versões das bibliotecas, armazenando os arquivos e registrando todas as mudanças realizadas e desta forma mantém um histórico das modificações que são informações extremamente importantes para o coordenador do projeto.

O controle de versões possui funções essenciais encontradas em controladores de

versões conhecidos, como por exemplo o Git ⁴ e Subversion ⁵, para gerenciar mudanças realizadas nas bibliotecas do repositório local. Para uma melhor organização, as versões das bibliotecas serão armazenadas em um diretório específico (por exemplo, VERSÕES) dentro do repositório local. É importante notar que a hierarquia criada para um projeto dentro do repositório principal também é criada dentro do repositório local (sob os diretórios USUÁRIO e VERSÕES), como mostrado na Figura 3.6. Essa replicação é necessária para compilação final do projeto.

Em relação à organização do sistema, cada projeto tem um coordenador responsável pela determinação de tarefas aos participantes. Com um comando de confirmação, um participante pode responder a uma tarefa, fazendo o *COMMIT*, por exemplo, de uma determinada biblioteca para o seu repositório local. Além disso, um segundo (terceiro e subsequentes) comando de *COMMIT*, antes de fazer o *upload* da biblioteca mais recente, o sistema cria uma versão da biblioteca atual armazenada no repositório local, copiando-a para o diretório destinado as versões da biblioteca corrente, de modo a realizar o controle de versão. O comando de *COMMIT*, portanto, irá:

- fornecer arquivos de bibliotecas funcionais estáveis para compilação.
- gerar *backups* (cópias) de bibliotecas, adicionando segurança ao projeto;
- permitir que os usuários restaurem mais tarde uma antiga biblioteca de trabalho, com um comando de *UPDATE*;

Após a conclusão das tarefas e submissão das bibliotecas pelos usuários (através do comando *COMMIT*), o coordenador pode realizar a compilação do projeto, se desejar. A compilação do projeto, realizada apenas pelo coordenador eleito, tem como objetivo copiar as bibliotecas dos repositórios locais e disponibilizá-las de maneira organizada no repositório principal. Outro processo realizado durante a compilação é a geração do arquivo "cde.lib" (responsável pelo armazenamento dos caminhos das bibliotecas instanciadas pelos projetistas), reunindo o conteúdo de todos os outros em apenas um arquivo. Em uma segunda compilação (terceira e demais), o sistema realiza apenas a cópia das bibliotecas que foram modificadas em seus repositórios locais, evitando que os arquivos sem alteração seja sobrescritos sem necessidade.

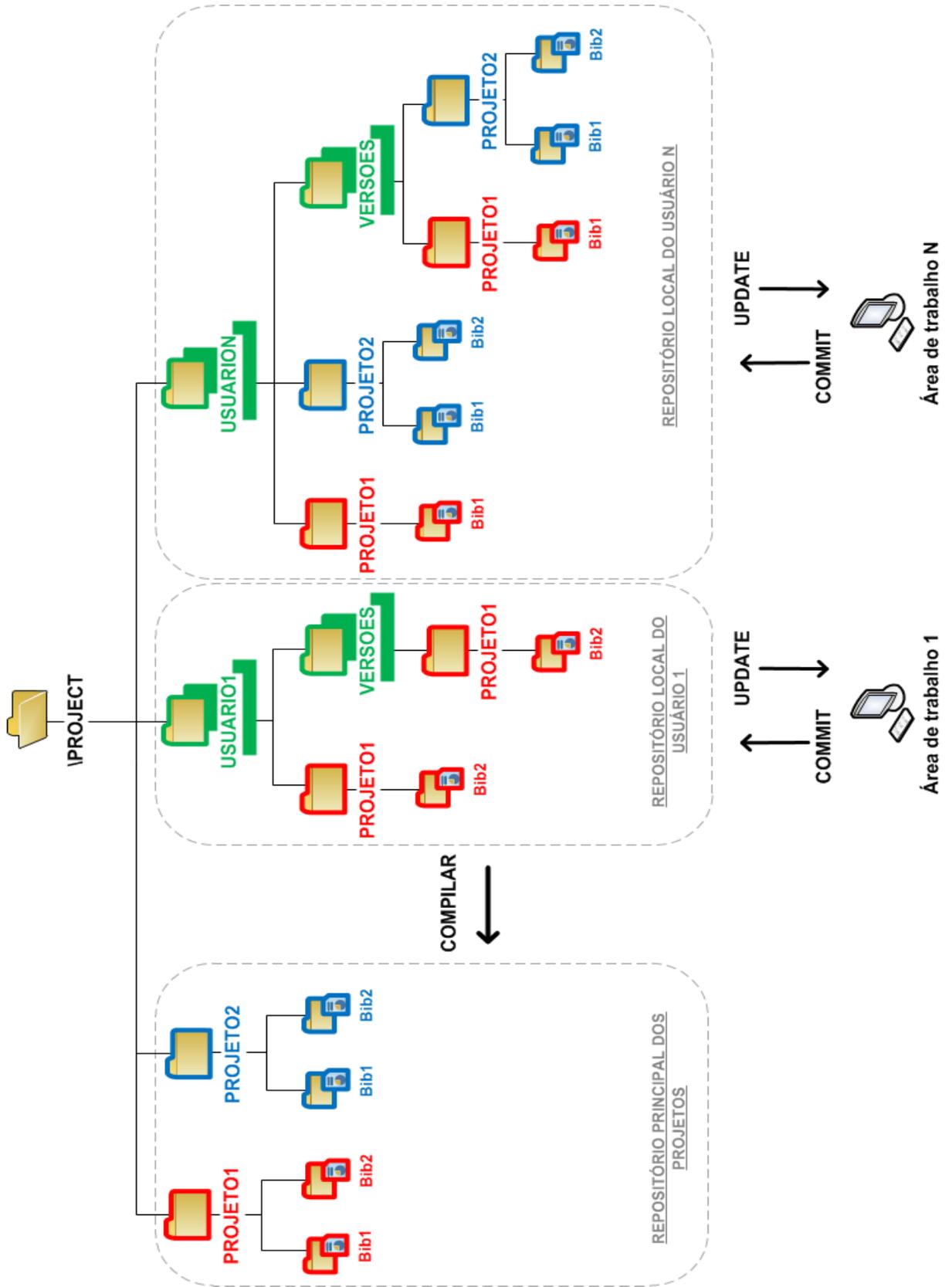
O coordenador do projeto também é responsável pela distribuição de tarefas, onde pode atribuir uma ou mais tarefas para uma determinada biblioteca. Em cada tarefa, o coordenador poderá definir qual sua prioridade (baixa, média ou alta) e prazo para conclusão.

Através dos componentes *Chat*, *E-mail* e Fórum, mensagens de texto são enviadas e recebidas para debaterem e decidirem sobre novas ideias ou problemas encontrados durante o trabalho colaborativo. Mensagens síncronas são trocadas pelo *Chat* e mensagens

⁴<<https://git-scm.com/>>

⁵<<https://subversion.apache.org/>>

Figura 3.6 – Hierarquia de diretórios do IC-Collab.



Fonte: Próprio Autor.

assíncronas pelo *E-mail*. O Fórum também permite a troca de mensagens assíncronas, porém sua principal função é promover um debate de determinada ideia ou problema a ser resolvido.

A Agenda permite que todos os participantes visualizem seus prazos para realizarem as tarefas definidas pelo coordenador através do componente Tarefas. Quando o participante é coordenador do projeto, além da visualização do prazo de suas tarefas, é facultado a ele visualizar os prazos dos demais participantes. Os demais participantes só podem visualizar os seus prazos. O Mural de recados serve para o coordenador incluir recados importantes aos participantes do projeto. O gerenciador de Usuários e gerenciador de Projetos também são acessíveis apenas pelo coordenador e possibilitam a inclusão de novas bibliotecas, adicionar novos participantes e gerenciar permissões de acesso nos diretórios do projeto.

Páginas e arquivos de ajuda podem ser criados e submetidos no ambiente através do componente *Wiki*. O *Wiki* provê um espaço contendo uma enciclopédia de páginas colaborativas com o objetivo de facilitar e agilizar o trabalho em projetos de CI.

Todas as atividades realizadas dentro de cada projeto serão registradas automaticamente pelo sistema e ficará disponível a todos os membros da equipe possibilitando que sejam gerados relatórios.

3.3 DESCRIÇÃO DO FUNCIONAMENTO DO IC-COLLAB

Após o desenvolvimento e validação do IC-COLLAB conforme determinam as fases do Ciclo de Desenvolvimento de *Groupware*, seguido de uma visão geral sistêmica, esta seção tem como objetivo descrever o funcionamento do ambiente virtual, abordando os requisitos de instalação, informações gerais disponíveis na tela inicial, tipos de perfis de usuário, importância do papel do coordenador dentro de um projeto, a realização do cadastro de usuários e de projetos, e o mais importante, como é realizado o trabalho colaborativo em equipe através do gerenciamento do projeto.

3.3.1 Requisitos para instalação

Para o funcionamento do IC-COLLAB é necessário um conjunto de tecnologias, as quais devem estar corretamente instaladas e configuradas. As tecnologias necessárias são listadas abaixo e o processo de instalação deve ser realizado conforme manual disponível no respectivo *site*:

- Sistema operacional Linux;
- Servidor *web* Apache 2.2 ou superior ⁶;

⁶Disponível em <<http://httpd.apache.org/docs/2.4/install.html>>

- Linguagem de programação PHP 5.3 ou superior ⁷;
- SGBD MySQL ou superior ⁸;
- Gerenciador *web* phpMyadmin 4.5 ou superior(para o MySQL) ⁹.

3.3.2 Tela inicial

A tela inicial do sistema, conforme ilustra a Figura 3.7, foi projetada e desenvolvida para mostrar informações gerais referentes aos projetos ao qual o usuário está participando e também sobre o desempenho do servidor. Uma lista informa quais os projetos do usuário e através dela é possível acessar o projeto. A agenda informa sobre o andamento do trabalho em modo geral, informando as tarefas em andamento e seus prazos para conclusão. Caso o usuário seja coordenador de um dos projetos, informações sobre atividades realizadas neste projeto são informadas com mais detalhes, por exemplo, data e hora do último *COMMIT* em determinada biblioteca. Um formulário de busca permite realizar pesquisas sobre os projetos existentes no sistema, possibilitando realizar filtros por título, categoria e participante. Informações relevantes sobre o servidor onde a *Cadence* está instalada, como o nome do sistema operacional e qual sua distribuição, bem como a versão da distribuição e quais usuários estão conectados remotamente no servidor. Informações de desempenho, como o uso do processador, uso da memória RAM e espaço em disco são mostradas através de gráficos com o objetivo de facilitar na solução de problemas de lentidão ou erros causados pelo uso excessivo durante as simulações realizadas na *Cadence*. No canto superior direito da página é exibido o nome do usuário que está acessando o sistema e ao clicar sobre ele, é possível alterar a senha de acesso (opção *Settings*) e também sair do sistema (opção *Logout*).

3.3.3 Perfis de usuário

Existem dois perfis de usuários no IC-COLLAB: ADMINISTRADOR e PARTICIPANTE. O perfil de ADMINISTRADOR possui todas as permissões dentro do ambiente, como a realização de cadastros e criação de projetos. O perfil de PARTICIPANTE possui permissão apenas dentro dos projetos onde participa, podendo realizar modificações conforme o desenvolvimento do trabalho. Caso o usuário seja eleito o coordenador de um projeto, ele terá permissão em todas as opções pertinentes ao projeto, como por exemplo, adicionar participantes, criar bibliotecas e tarefas, etc. É importante esclarecer que quando um usuário com perfil de ADMINISTRADOR está participando de um projeto e não é o coordenador do mesmo, ele não possui os acessos de coordenador.

⁷Disponível em <https://secure.php.net/manual/pt_BR/install.php>

⁸Disponível em <<https://dev.mysql.com/doc/refman/8.0/en/installing.html>>

⁹Disponível em <https://docs.phpmyadmin.net/pt_BR/latest/setup.html>

Figura 3.7 – Tela inicial do IC-COLLAB.

Search projects

Search for...

My Projects

Calendar

Show **10** entries

Search:

| ID | Title | Category | Coordinator | Created by | Date creation | Action |
|----|----------------|-----------------|------------------|------------------|---------------------|--------|
| 2 | OSCRELAX | ANALOG CIRCUITS | ANDRE AITA | ALDAMI WONTROBA | 06/06/2017 10:33:13 | |
| 4 | RINGOSCILLATOR | ANALOG CIRCUITS | ANDRE AITA | ANDRE AITA | 03/10/2017 10:44:05 | |
| 9 | Projeto1 | OTHERS | PETER PREVEDELLO | PETER PREVEDELLO | 07/03/2018 15:14:04 | |

Showing 1 to 3 of 3 entries

<
>
today
month
week
day

March 2018

| Sun | Mon | Tue | Wed | Thu | Fri | Sat |
|-----|-----|-----|-----|-----|-----|-----|
| 25 | 26 | 27 | 28 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| 18 | 19 | 20 | 21 | 22 | 23 | 24 |

Server

Operational system:
Linux

Distribution:
CentOS

Version:
6.2

Users Logged in to SERVER:
felipe_a
padilha
padilha
talesibd

CPU

RAM MEMORY

DISK

Fonte: Próprio Autor.

Figura 3.8 – Tela de cadastro de usuários no IC-COLLAB.

The screenshot displays the 'Users' management interface. On the left is a sidebar with navigation links. The main area shows a table of users with the following data:

| ID | Name | E-mail | Profile | Actions |
|----|------------------|-----------------------------|---------------|---------|
| 1 | ADMINISTRADOR | admin@admin | ADMINISTRADOR | [Edit] |
| 14 | PETER PREVEDELLO | peterprevedello@gmail.com | ADMINISTRADOR | [Edit] |
| 15 | ANDRE AITA | aaita@inf.ufsm.br | ADMINISTRADOR | [Edit] |
| 16 | ALDAIR WONTROBA | aldair.wontroba@hotmail.com | PARTICIPANTE | [Edit] |
| 17 | TALES BORTOLIN | talesbd@gmail.com | ADMINISTRADOR | [Edit] |

At the bottom of the table, it indicates 'Showing 1 to 5 of 5 entries' and includes 'Previous' and 'Next' navigation buttons.

Fonte: Próprio Autor.

3.3.4 Cadastro de usuários

Disponível apenas para usuários com perfil de ADMINISTRADOR, os cadastros são requisitos básicos para o funcionamento do sistema. Ao utilizar o IC-COLLAB pela primeira vez, deve-se primeiro cadastrar os usuários (através da opção "Users") que irão utilizá-lo. Por meio da interface ilustrada na Figura 3.8, é possível adicionar novos usuários e editar suas informações sempre que necessário.

No momento da criação do usuário, algumas informações são solicitadas e dentre elas, as mais importantes são as regras no campo *login* de acesso e a definição do perfil. O nome do *login* de acesso deve ser idêntico ao nome da sessão que o usuário utiliza para acessar remotamente (via protocolo SSH) o servidor (Centos) onde a *Cadence está instalada*, pois é através dessa igualdade que o IC-COLLAB criará um vínculo do repositório local com a área de trabalho local do usuário. Essa regra se faz necessária porque ao finalizar o cadastro do novo usuário, um diretório (repositório local) com o mesmo nome do *login* de acesso é criado no ambiente, por exemplo o "USUARIO1" na Figura 3.6. Ao criar o diretório com mesmo nome do *login*, também é criado um subdiretório, denominado "VERSOES", para realizar o controle de versões dos trabalhos realizados pelo usuário. Uma vez criado o usuário com determinado nome de *login*, não poderá ser alterado e nem se repetir dentro do ambiente. A definição do perfil do usuário também é importante para realizar o controle de acessos, pois conforme descrevemos na seção anterior, o perfil ADMINISTRADOR possui todas as permissões dentro do ambiente.

3.3.5 Coordenador de projeto

Antes de descrever sobre o processo de cadastro de projetos, é fundamental entender sobre a função do coordenador no IC-COLLAB. O coordenador de projetos no IC-COLLAB, na prática, poderá exercer tanto o papel de líder como o de mediador dos conhecimentos. O papel de líder possui responsabilidade de planejar e controlar a execução do projeto de tal forma que os participantes não entrem em conflitos e estejam sempre motivados e sendo avaliados sobre desempenho e prazos durante a realização das suas atividades (PMI, 2008). Um professor, além de ser educador e transmissor de conhecimento, ao mesmo tempo atua como mediador, ou seja, o professor deve se colocar como ponte entre o estudante e o conhecimento para que, dessa forma, o aluno aprenda a "pensar" e a questionar por si mesmo e não mais receba passivamente as informações como se fosse um depósito do educador (BULGRAEN, 2010). Desta forma, no IC-COLLAB, o coordenador é o responsável pela organização e gestão dos trabalhos dos demais participantes durante as fases de projeto de CIs. Além disso, o ambiente IC-COLLAB permite que o coordenador também atue eventualmente como participante do projeto, desenvolvendo uma ou mais bibliotecas.

3.3.6 Cadastro de projetos

Assim como no cadastro de usuários, os projetos também só podem ser criados por usuários com perfil de ADMINISTRADOR. Ao adicionar um novo projeto, é necessário obedecer uma sequência de 4(quatro) passos. Esse processo se faz necessário para que a hierarquia de diretórios, tanto no repositório principal quanto no repositório local, seja criada corretamente e de maneira organizada. A seguir, o processo de cadastro será detalhado passo a passo e em seguida através de um fluxograma de processo.

Passo 1: Informações do projeto

O primeiro passo, como podemos ver na Figura 3.9, é destinado ao cadastro de informações do projeto como categoria, título e uma breve descrição. A categoria deve ser preenchida conforme o tipo do projeto, podendo ele ser por exemplo, um circuito digital, um circuito analógico ou ambos. Assim como no cadastro de usuários, o título do projeto deve respeitar a mesma regra do campo *login* de acesso, onde é permitido informar apenas letras e números e não símbolos ou caracteres especiais. O título não pode se repetir em outro projeto e é ele que define o nome do diretório do projeto no repositório principal, por exemplo o "PROJETO1" da Figura 3.6.

Passo 2: Adicionar participantes ao projeto

No segundo passo devem ser adicionados os usuários que irão participar e colaborar no projeto. Conforme a Figura 3.10, uma interface facilita o processo listando

Figura 3.9 – Passo 1 do cadastro de projeto: cadastrar informações.

NEW PROJECT

Step 1 of 4

Project | Participants | Coordinator | Libraries

Category
DIGITAL CIRCUITS

Title
Projeto2

Description
O Projeto2 tem será utilizado para testes no IC-COLLAB.

Save and Continue | Clean | Close

Fonte: Próprio Autor.

abaixo todos os usuários cadastrados no sistema, bastando apenas adicioná-los ao projeto clicando no botão "+ Add". Ao adicionar um usuário como participante do projeto, o sistema cria um diretório (com o mesmo nome dado projeto) no repositório local, assim como no "USUARIO1", exemplificado na Figura 3.6. No caso da adição equivocada de um usuário, o processo é desfeito deletando-o.

Passo 3: Definir um participante como coordenador do projeto

Um participante deve ser definido como coordenador do projeto no passo três. Para um melhor controle de acesso ao diretório do projeto no repositório principal, cada projeto poderá ter apenas um coordenador com todas as permissões (leitura, escrita e execução). O controle de permissões é realizado via controle de permissões¹⁰ do sistema operacional *Centos* e são concedidas no momento que o participante é definido como coordenador. Os demais participantes terão apenas permissão de leitura no diretório, ou seja, não poderão realizar nenhuma modificação nos arquivos. Como podemos ver na Figura 3.11, todos os participantes adicionados no passo anterior são listados e para definir um deles como coordenador, basta apenas clicar no botão "+ Coordinator".

Passo 4 Criar bibliotecas para os participantes

No passo quatro deve-se criar as bibliotecas para os participantes. Uma biblioteca será atribuída a apenas um participante, porém um participante poderá trabalhar em uma ou mais bibliotecas. É importante informar que o coordenador é um participante do projeto como os demais e poderá também trabalhar em uma ou mais bibliotecas, conforme mostra a Figura 3.12. Ao criar uma biblioteca, primeiramente é necessário informar o participante que trabalhará nela, em seguida será preciso informar um título para a mesma, que assim como no cadastro de usuários (*login* de acesso) e no título do projeto, é permitido informar apenas letras e números e não símbolos ou caracteres especiais, pois é ele que define o nome do diretório da biblioteca. Esse diretório será criado um nível abaixo do diretório do projeto no repositório principal, por exemplo a "Bib1" do "PROJETO1" da Figura 3.6. Da mesma forma, um diretório de mesmo nome será criado no repositório local do usuário, mantendo a mesma estrutura hierárquica, como por exemplo a "Bib1" do "PROJETO1" do "USUARIO1", exemplificado na Figura 3.6. Em seguida, basta finalizar o cadastro clicando no botão "Finish registration".

A Figura 3.13, ilustra através de um fluxograma, a sequência de passos para criar um projeto no IC-COLLAB. Fluxogramas são formas de representar, por meio de símbolos gráficos, o passo a passo de um processo e sem dúvida também facilita o seu entendimento (PEINADO; GRAEML, 2007).

¹⁰Disponível em <<https://www.centos.org/docs//2/rhl-gsg-en-7.2/s1-navigating-ownership.html>>

Figura 3.10 – Passo 2 do cadastro de projeto: adicionar participantes.



Projeto2

Step 2 of 4

Project
 Participants
 Coordinator
 Libraries

Project Participants

Show entries
Search:

| ID | Name | Action |
|----|------------------|---|
| 14 | PETER PREVEDELLO | To remove |

Showing 1 to 1 of 1 entries
Previous Next

+ Add Participant

Show entries
Search:

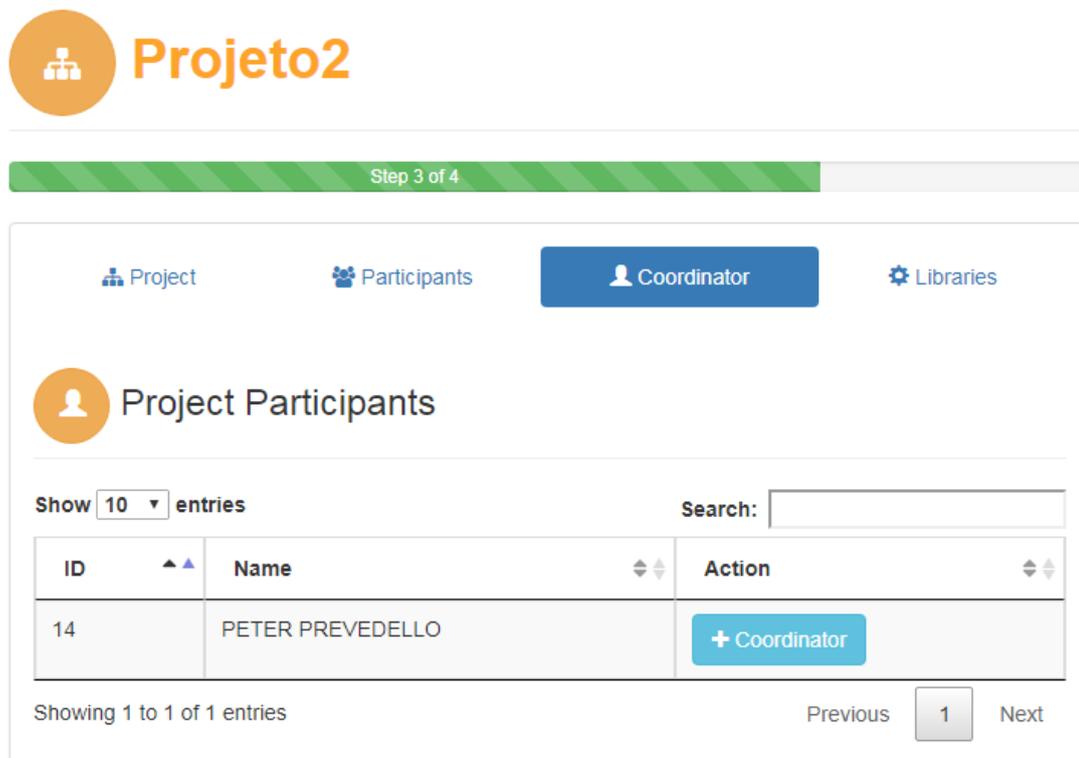
| ID | Name | Action |
|----|-----------------|--|
| 15 | ANDRE AITA | + Add |
| 16 | ALDAIR WONTROBA | + Add |
| 17 | TALES BORTOLIN | + Add |

Showing 1 to 3 of 3 entries
Previous Next

Continue Registration

Fonte: Próprio Autor.

Figura 3.11 – Passo 3 do cadastro de projeto: Definir um participante como coordenador.



Fonte: Próprio Autor.

3.3.7 Gerenciamento do projeto

O gerenciamento do projeto é realizado através de um conjunto de funcionalidades disponíveis em apenas uma tela do IC-COLLAB e, conforme ilustra a Figura 3.14, estão disponíveis lado a lado para melhor facilitar a usabilidade do ambiente pelos usuários. As funcionalidades *Project*, *Participants*, *Libraries*, *Versions*, *Tasks*, *Wiki*, *Historic*, *Forum* e *cds.lib* são detalhados a seguir:

- *Project*

Na opção "*Project*" são visualizados os seguintes campos contendo informações do projeto: título, descrição, caminho do repositório central e categoria. O sistema permite somente a alteração dos campos descrição e categoria, os demais campos não podem ser alterados por possuírem vínculo com a estrutura hierárquica de diretórios. O botão "*Compile MAIN REPOSITORY*" efetua a compilação (cópia da última versão) das bibliotecas submetidas ("*COMMIT*") pelos participantes (armazenadas nos repositórios locais), para o repositório central. Já o botão "*Manage MAIN REPOSITORY*" permite ao coordenador navegar pelo repositório principal do projeto para visualizar detalhes como, data e hora de modificação dos diretórios e arquivos compilados. Todas essas opções estão disponíveis somente ao coordenador do projeto.

- *Participants*

Figura 3.12 – Passo 4 do cadastro de projeto: Criar bibliotecas para os participantes.

Projeto2

Step 4 of 4

Project Participants Coordinator Libraries

New LIBRARY

Libraries

Show 10 entries Search:

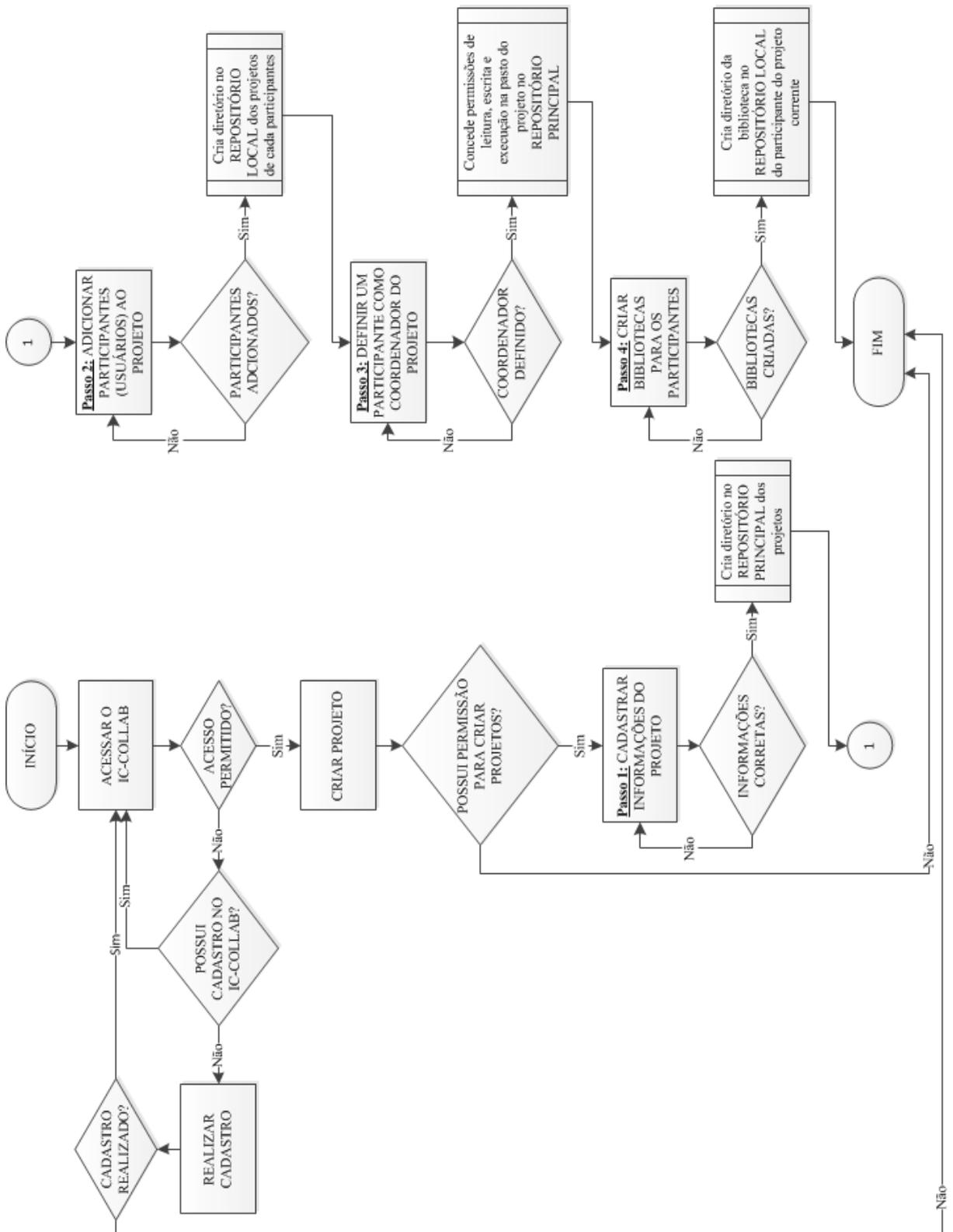
| ID | Title | Responsible | Action |
|----|-------|------------------|--------|
| 15 | Bib1 | PETER PREVEDELLO | |
| 16 | Bib2 | PETER PREVEDELLO | |

Showing 1 to 2 of 2 entries Previous 1 Next

Finish registration

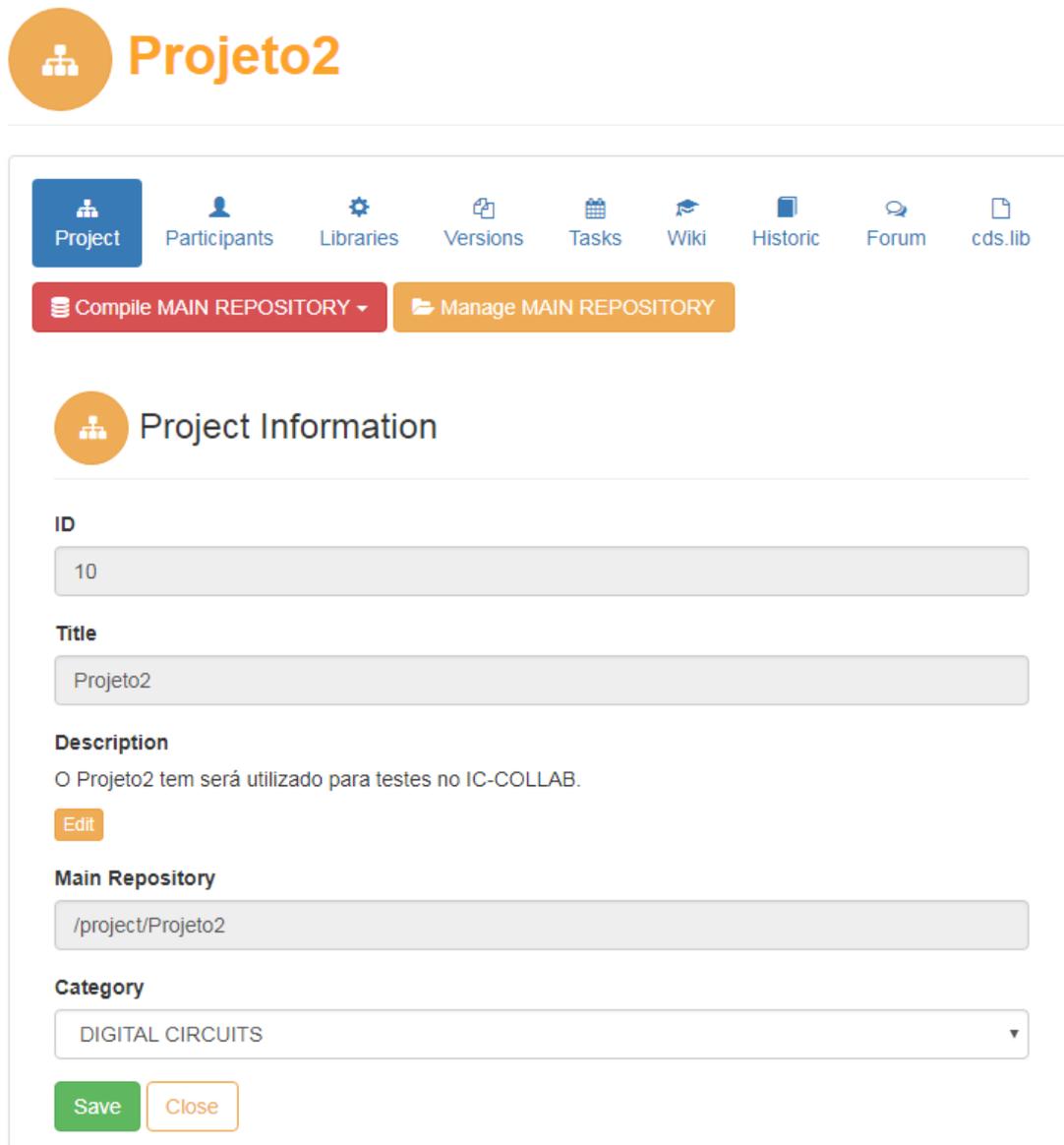
Fonte: Próprio Autor.

Figura 3.13 – Fluxograma do processo de criação de projetos no IC-COLLAB.



Fonte: Próprio autor.

Figura 3.14 – Gerenciamento do projeto.



The image shows a web interface for managing a project named 'Projeto2'. At the top, there is a navigation bar with icons for Project, Participants, Libraries, Versions, Tasks, Wiki, Historic, Forum, and cds.lib. Below this, there are two buttons: 'Compile MAIN REPOSITORY' and 'Manage MAIN REPOSITORY'. The main content area is titled 'Project Information' and contains several fields:

- ID:** A text input field containing the value '10'.
- Title:** A text input field containing the value 'Projeto2'.
- Description:** A text area containing the text 'O Projeto2 tem será utilizado para testes no IC-COLLAB.' Below this is an 'Edit' button.
- Main Repository:** A text input field containing the value '/project/Projeto2'.
- Category:** A dropdown menu with the selected value 'DIGITAL CIRCUITS'.

At the bottom of the form, there are two buttons: 'Save' and 'Close'.

Fonte: Próprio Autor.

Todos os participantes do projeto são visualizados na opção "*Participants*". Adicionar e/ou remover participantes, bem como alterar quem é o coordenador do projeto podem ser realizados nessa opção e para tal, somente o atual coordenador tem acesso.

- *Libraries*

Todas as bibliotecas podem ser visualizadas na opção "*Libraries*". As bibliotecas são visualizadas de duas formas: Minhas Bibliotecas ("*My Libraries*") e Outras Bibliotecas ("*Other Libraries*"). As Minhas Bibliotecas, como o próprio nome diz, são listadas as bibliotecas pertencentes ao participante que está acessando o sistema e somente ele tem acesso a realizar operações de "*Commit*" e "*Update*". A operação de "*Commit*" refere-se ao processo de submissão das últimas alterações realizadas em uma biblioteca ao repositório local do participante. Cada vez que essa operação é realizada, o sistema gera uma nova versão da biblioteca, salvando os dados anteriores a submissão das últimas alterações e ficam disponíveis na opção "*Versions*". Mais tarde, quando a biblioteca for concluída, ela fica disponível para o coordenador realizar a compilação para o repositório principal do projeto. O "*Commit*" pode ser realizada uma ou várias vezes e a qualquer momento uma versão mais antiga pode ser restaurada, caso algum problema ou erro ocorra. A operação de "*Update*" possibilita realizar o processo inverso do "*Commit*", baixando os dados armazenados no repositório local do participante para sua área de trabalho local. As bibliotecas dos demais participantes do projeto são listadas em Outras Bibliotecas e não é possível realizar nenhum tipo de operação ("*Commit*" e "*Update*"), apenas é permitida a visualização dos dados armazenados na biblioteca.

- *Versions*

As versões das bibliotecas geradas anteriormente são listadas na opção "*Versions*". Da mesma forma que na opção *Libraries*, as versões são listadas de duas formas: Minhas versões ("*My Versions*") e Outras Versões ("*Other Versions*"). Versões pertencentes às bibliotecas do participante que está acessando o sistema são listadas em Minhas Versões e as versões das bibliotecas dos demais participantes são listadas em Outras Versões. Essa funcionalidade foi criada para informar aos participantes as versões das bibliotecas e também a data e hora de submissão.

- *Tasks*

O coordenador do projeto (e somente ele) é responsável pela atribuição de tarefas aos participantes. Essa funcionalidade é disponibilizada na opção "*Tasks*" e é necessário que o participante seja responsável por uma biblioteca, pois a tarefa é compulsoriamente vinculada a ela. Ao criar uma tarefa, algumas informações são solicitadas, como o nome da biblioteca, título, descrição e prazo para a sua conclusão.

Cada participante deve adicionar acompanhamentos em suas tarefas, informando o andamento do seu trabalho.

- *Wiki*

A opção "*Wiki*" possibilita aos participantes criarem e visualizarem páginas com materiais de ajuda. Essas páginas são compartilhadas com todos os usuários do IC-COLLAB e podem ser revisadas sempre que um participante julgar necessário. Ao criar uma página, os seguintes campos devem ser preenchidos: título, descrição e arquivo. Os campos título e descrição são obrigatórios, pois devem ser preenchidos com informações úteis aos demais colegas. Arquivos com tutoriais e/ou referências sobre determinado assunto podem ser anexados no campo arquivo.

- *Historic*

Disponível a toda equipe em um projeto, na opção "*Historic*" ficam registradas todas as atividades realizadas dentro do projeto. Disponível a todos os participantes, as informações listadas nessa funcionalidade servem de apoio ao coordenador do projeto, uma vez que provê todo o histórico de atividades de determinado usuário, o que é fundamental para a coordenação efetiva. Além disso, o histórico de atividades auxilia o administrador do sistema no encaminhamento de solução de falhas técnicas que poderão vir a ocorrer.

- *cds.lib*

A opção "*cds.lib*" apresenta o conteúdo do arquivo *cds.lib* do projeto, criado no momento da primeira compilação das bibliotecas. O propósito é evidentemente a conveniência, uma vez que o arquivo é gerado a partir das bibliotecas dos participantes. Evidente também é a necessidade de se adicionar os caminhos das bibliotecas utilizadas pelos *design kits* e tecnologias em uso pelo ambiente *Cadence*. Para o coordenador do projeto é disponibilizada a opção de editar o arquivo através de um formulário, uma vez que ajustes normalmente são necessários, possibilitando adicionar manualmente caminhos de outras bibliotecas. Aos demais participantes é permitido apenas a visualização do conteúdo do arquivo *cds.lib*.

4 ESTUDOS DE CASO

Para demonstrar a integração das tecnologias utilizadas, bem como apresentar o ambiente virtual IC-COLLAB e comprovar seus benefícios, foram realizados dois estudos de caso em um dos laboratórios do Grupo de Microeletrônica (GMICRO) da Universidade Federal de Santa Maria (UFSM).

No primeiro caso foi proposto um projeto mais simples, onde foi construído um Oscilador em Anel. Já no segundo caso, um projeto maior e mais complexo foi proposto para a construção de um Coletor de Energia Piezoelétrico. Ambos os casos envolveram uma equipe de três participantes, incluindo o coordenador.

4.1 ESTUDO DE CASO 1

O Oscilador em Anel é um circuito integrado muito utilizado para a determinação dos tempos de propagação de uma porta inversora em determinada tecnologia. Constitui-se de N (sendo N um número ímpar) inversores montados em cascata, em que a saída do último inversor é realimentada para a entrada do primeiro (KERBER et al., 2017).

Neste estudo de caso, o Oscilador em Anel foi projetado utilizando três inversores ($N=3$), onde cada inversor é considerado uma biblioteca dentro do ambiente IC-COLLAB, desenvolvidas por diferentes projetistas e seguiu da seguinte forma.

4.1.1 Cadastro de Usuários

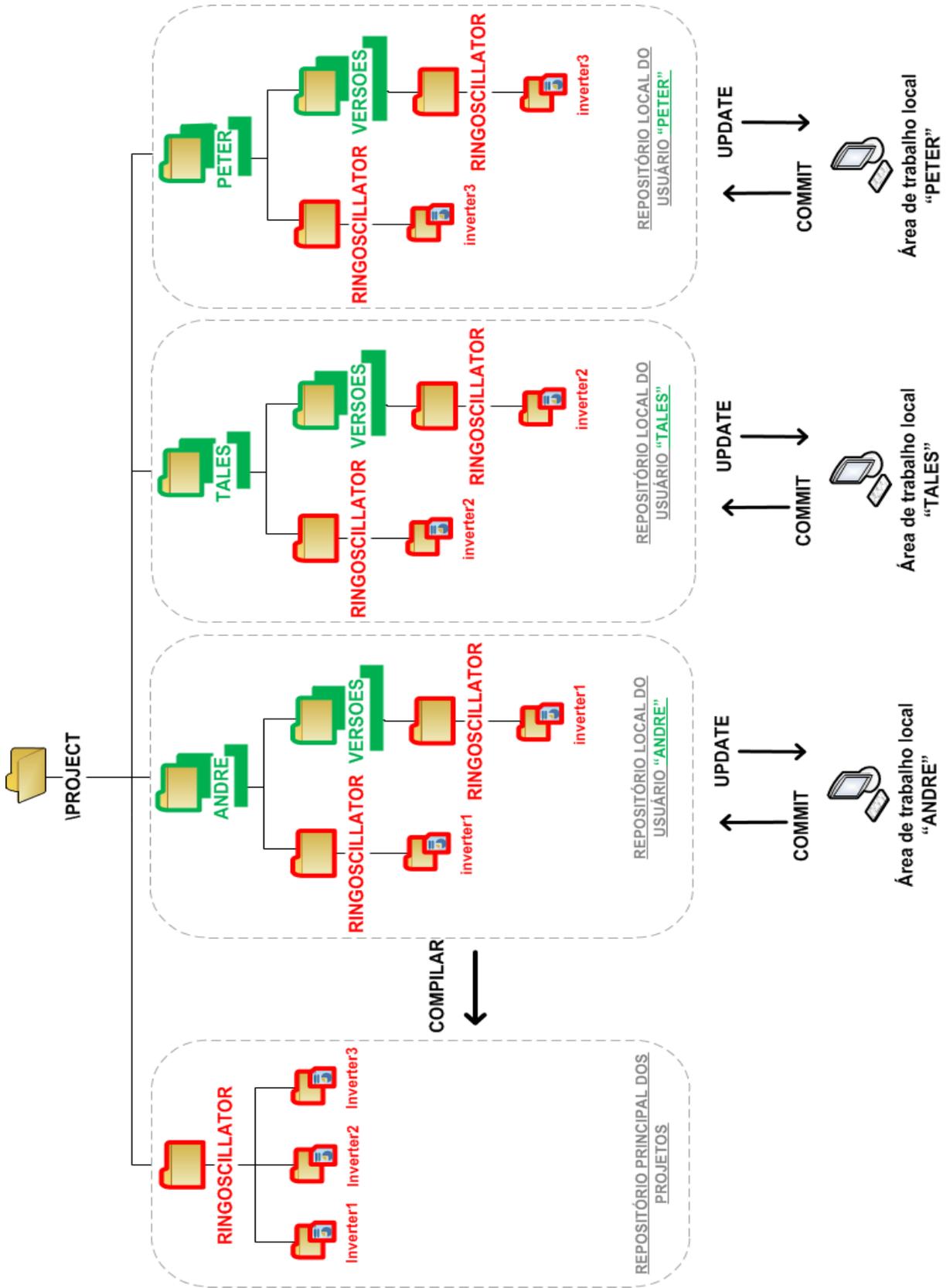
Primeiro, os usuários foram registrados no sistema IC-COLLAB. Para cada usuário registrado, um repositório local de usuário foi criado pelo sistema (pastas ANDRE, TALES e PETER, conforme apresenta a Figura 4.1). É importante saber que o registro de um usuário é realizado uma única vez no sistema e portanto, não se repetirá mais.

4.1.2 Cadastro do projeto

Em sequência, o projeto foi registrado através de um assistente que divide o processo em 4(quatro) passos. No primeiro passo, o título foi definido, bem como sua categoria e descrição (como ilustra a Figura 4.2). O título do projeto, neste estudo de caso RINGOSCILLATOR, define o nome da pasta do repositório principal e, por isso, precisa ser único no ambiente.

No passo seguinte, os usuários foram adicionados como participantes do projeto. Ainda nesta etapa, após a adição de um usuário é criada a pasta RINGOSCILLATOR no repositório local de cada usuário, replicando a estrutura do repositório principal do projeto, conforme Figura 4.1.

Figura 4.1 – Hierarquia de diretórios do projeto RINGOSCILATOR.



Fonte: Próprio Autor.

Figura 4.2 – Informações do projeto RINGOSCILLATOR.

RINGOSCILLATOR

Project Participants Libraries Versions Tasks Wiki Historic Forum cds.lib

Compile MAIN REPOSITORY Manage MAIN REPOSITORY

Project Information

ID
4

Title
RINGOSCILLATOR

Description
Ring oscillator with five inverters, used as a case study for ISCAS2018, Florence, Italy.
Edit

Main Repository
/project/ringoscillator

Category
ANALOG CIRCUITS

Save Close

Fonte: Próprio Autor.

No terceiro passo, um dos participantes foi designado ou eleito como coordenador do projeto e, portanto, ficando como responsável pela definição das tarefas e do gerenciamento de todas as funções do projeto (incluindo a compilação).

No último passo, as bibliotecas foram criadas (inverter1, inverter2 e inverter3), definindo título, descrição e o participante responsável. Vale ressaltar que as bibliotecas precisam ser definidas com base em um estudo prévio de diagrama de blocos de projeto/circuito, com responsabilidades associadas aos participantes. No presente estudo de caso, cada um dos participantes foi associado a uma biblioteca, inclusive o coordenador, pois ele também pode atuar como participante. Do mesmo modo que no primeiro e segundo passo, ao criar uma biblioteca, uma pasta com mesmo nome do título dado a ela é criada no repositório principal e também no repositório local do usuário responsável, como apresenta a Figura 4.1.

Figura 4.3 – Bibliotecas criadas no projeto RINGOSCILATOR.

RINGOSCILLATOR

Project Participants **Libraries** Versions Tasks Wiki Historic Forum cds.lib

New LIBRARY

My Libraries

Show 10 entries Search:

| ID | Title | Responsible | Actions |
|----|-----------|-------------|---------|
| 4 | INVERTER1 | ANDRE AITA | |

Showing 1 to 1 of 1 entries Previous 1 Next

Other Libraries

Show 10 entries Search:

| ID | Title | Responsible | Actions |
|----|-----------|------------------|---------|
| 5 | INVERTER2 | TALES BORTOLIN | |
| 7 | INVERTER3 | PETER PREVEDELLO | |

Showing 1 to 2 of 2 entries Previous 1 Next

Fonte: Próprio Autor.

4.1.3 Projeto Colaborativo

O coordenador do projeto possui opções e permissões diferentes no projeto e portanto, definiu tarefas com prazo estipulado para os demais participantes. Cada participante precisa adicionar acompanhamentos descrevendo detalhadamente o que está sendo realizado para resolver sua tarefa, de modo a manter o coordenador bem informado. Reuniões e discussões sobre determinado assunto envolvendo o projeto foram realizadas utilizando os componentes *e-mail*, *chat* ou Fórum. O compartilhamento de materiais de ajuda utilizando o componente *wiki* também foi realizado.

Depois que as bibliotecas foram finalizadas, (esquemáticos simulados com símbo-

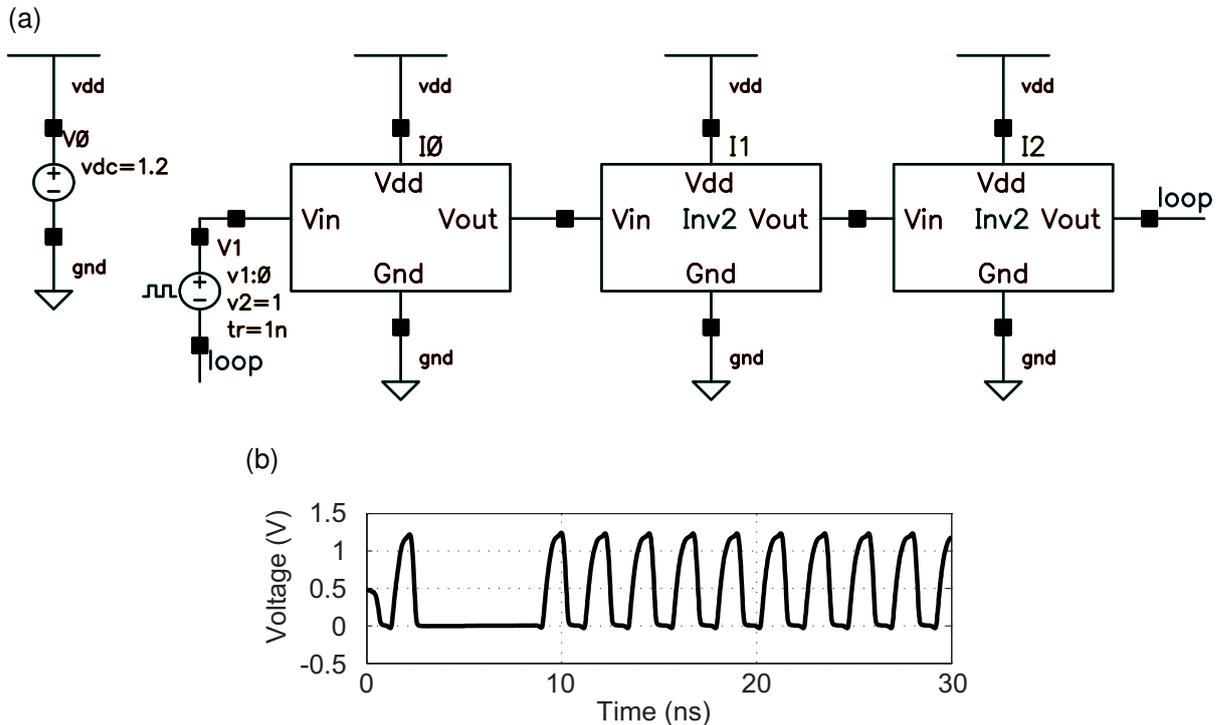


Figura 4.4 – (a) Esquemático do Oscilador em Anel após a compilação, e (b) simulação elétrica.

los e pinos associados e eventualmente *layout*), os participantes submeteram suas bibliotecas (esta operação foi realizada através do botão "COMMIT", visível na Figura 4.3), duas vezes, a verificar o "UPLOAD" e o controle de versão da biblioteca (em seus respectivos repositórios locais).

4.1.3.1 Compilação e simulação

Com os repositórios locais preenchidos e atualizados, o coordenador compilou (via botão "Compile MAIN REPOSITORY", visível na Figura 4.2) o projeto, copiando as bibliotecas dos repositórios locais dos participantes para o repositório principal, tendo todas as bibliotecas do projeto RINGOSCILLATOR disponíveis para finalizar o projeto. A visualização das bibliotecas (arquivos e pastas) compiladas é possível através do botão "Manage MAIN REPOSITORY"). Se o coordenador tivesse encontrado alguma biblioteca com problema, ele poderia ter enviado uma mensagem para o aluno responsável (via *e-mail* ou *chat*), com instruções apropriadas.

A Figura 4.4 apresenta o esquemático do Oscilador em Anel (com a simulação elétrica em anexo), após a compilação no repositório principal.

4.2 ESTUDO DE CASO 2

O Coletor de Energia Piezoelétrico é um circuito integrado que converte a energia cinética do movimento em energia elétrica, que ao passar por um circuito coletor, é retificada e amplificada, desta forma possibilita que a energia seja utilizada ou armazenada em uma bateria ou capacitor (KWON; RINCÓN-MORA, 2014) (MALVINO; BATES, 2016). Vale ressaltar que este projeto possui maior complexidade que o projeto do estudo de caso anterior (Oscilador em Anel), possuindo também maior número de bibliotecas, divididas entre os participantes.

4.2.1 Cadastro do projeto

Assim como no primeiro estudo de caso, um novo projeto foi registrado no IC-COLLAB. Utilizando o assistente em quatro passos do ambiente, no primeiro passo foi definido o título, a categoria e a descrição do projeto, conforme apresenta a Figura 4.5. Neste segundo estudo de caso, o título foi definido como PiezoEnergy, que também define o nome da pasta do repositório principal e portanto, é único no ambiente.

Em seguida, no segundo passo foram adicionados os usuários participantes do projeto, onde a pasta PiezoEnergy foi criada no repositório local de cada usuário, replicando a estrutura do repositório principal do projeto. Ainda nesta etapa, o ambiente IC-COLLAB também dispara um *e-mail* para o usuário, informando-o sobre sua participação no novo projeto.

No passo seguinte (terceiro passo), um dos participantes foi definido como coordenador do projeto e, portanto, ficando como responsável pela atribuição de tarefas aos demais participantes e também pelo gerenciamento de todas as funções do projeto.

No último passo, as bibliotecas foram criadas definindo título, descrição e o participante responsável, conforme apresenta a Figura 4.6.

Do mesmo modo que no primeiro e segundo passo, ao criar uma biblioteca, uma pasta com mesmo nome do título dado a ela é criada no repositório principal e também no repositório local do usuário responsável.

4.2.2 Projeto Colaborativo

Após a criação das bibliotecas e definição dos seus responsáveis, o coordenador definiu tarefas com prazos para entrega. Durante o desenvolvimento das bibliotecas, os participantes descreviam sobre o que estava sendo realizado, adicionando acompanhamentos em suas tarefas e desta forma mantendo toda a equipe informada sobre o andamento do trabalho.

A medida que cada participante finalizava suas atividades referente as suas respectivas bibliotecas, também era realizado o "*COMMIT*" de cada uma delas, enviando-as para

Figura 4.5 – Informações do projeto PiezoEnergy.

The screenshot shows a web interface for the PiezoEnergy project. At the top, there is a navigation bar with icons for Project, Participants, Libraries, Versions, Tasks, Wiki, Historic, Forum, and cds.lib. Below the navigation bar, there are two buttons: 'Compile MAIN REPOSITORY' (red) and 'Manage MAIN REPOSITORY' (orange). The main section is titled 'Project Information' and contains the following fields:

- ID:** 7
- Title:** PiezoEnergy
- Description:** Coletor de energia por meio de um piezo. There is an 'Edit' button next to this field.
- Main Repository:** /project/PiezoEnergy
- Category:** ANALOG CIRCUITS (dropdown menu)

At the bottom of the form, there are 'Save' and 'Close' buttons.

Fonte: Próprio Autor.

o seu repositório local. Para cada biblioteca, foi realizado dois ou mais "COMMIT".

O compartilhamento de materiais de ajuda e manuais foi realizado utilizando o componente *Wiki*, conforme apresentado na Figura 4.7. O *Wiki* possibilita a edição colaborativa do material postado por um outro participante, mesmo que ele não seja o autor do material.

4.2.2.1 Compilação e simulação

Por fim, o coordenador compilou (via botão "*Compile MAIN REPOSITORY*", visível na Figura 4.5) o projeto, agrupando todas as bibliotecas dos repositórios locais dos participantes no repositório principal, tendo todas as bibliotecas do projeto PiezoEnergy disponíveis para finalizar o projeto. Para confirmar que todas as bibliotecas encontravam-se no repositório principal, o coordenador utilizou o botão "*Manage MAIN REPOSITORY*"), que possibilita a visualização dos arquivos e pastas que compõem as bibliotecas. Durante toda construção do projeto, informações eram trocadas através de *e-mail* e/ou *chat*. A Figura

Figura 4.6 – Bibliotecas criadas no projeto PiezoEnergy.

 **PiezoEnergy**

Project Participants Libraries Versions Tasks Wiki Historic Forum cds.lib

[New LIBRARY](#)

My Libraries

Search:

| ID | Title | Responsible | Actions |
|----|--------------------------|----------------|--|
| 12 | SwitchAnalog | TALES BORTOLIN |       |
| 18 | PiezoHarvesting | TALES BORTOLIN |       |
| 19 | LogicControl | TALES BORTOLIN |       |
| 20 | Switches | TALES BORTOLIN |       |
| 21 | PeakDetector | TALES BORTOLIN |       |
| 22 | CompCrossZero | TALES BORTOLIN |       |
| 23 | InvestControl | TALES BORTOLIN |       |
| 25 | ComparatorEnableVerilogA | TALES BORTOLIN |       |
| 26 | ModelDeGlith | TALES BORTOLIN |       |

Showing 1 to 9 of 9 entries

Other Libraries

Search:

| ID | Title | Responsible | Actions |
|----|---------------------|------------------|---|
| 13 | ModelLEnergyDrain | ANDRE AITA |     |
| 24 | PiezoTransducer | PETER PREVEDELLO |     |
| 27 | ModelLEnergyDrainTB | ANDRE AITA |     |

Showing 1 to 3 of 3 entries

Fonte: Próprio Autor.

Figura 4.7 – Compartilhamento de materiais no componente *Wiki* do ambiente IC-COLLAB.

PiezoEnergy

Project Participants Libraries Versions Tasks **Wiki** Historic Forum cds.lib

New TUTORIAL

Support materials

Show 10 entries Search:

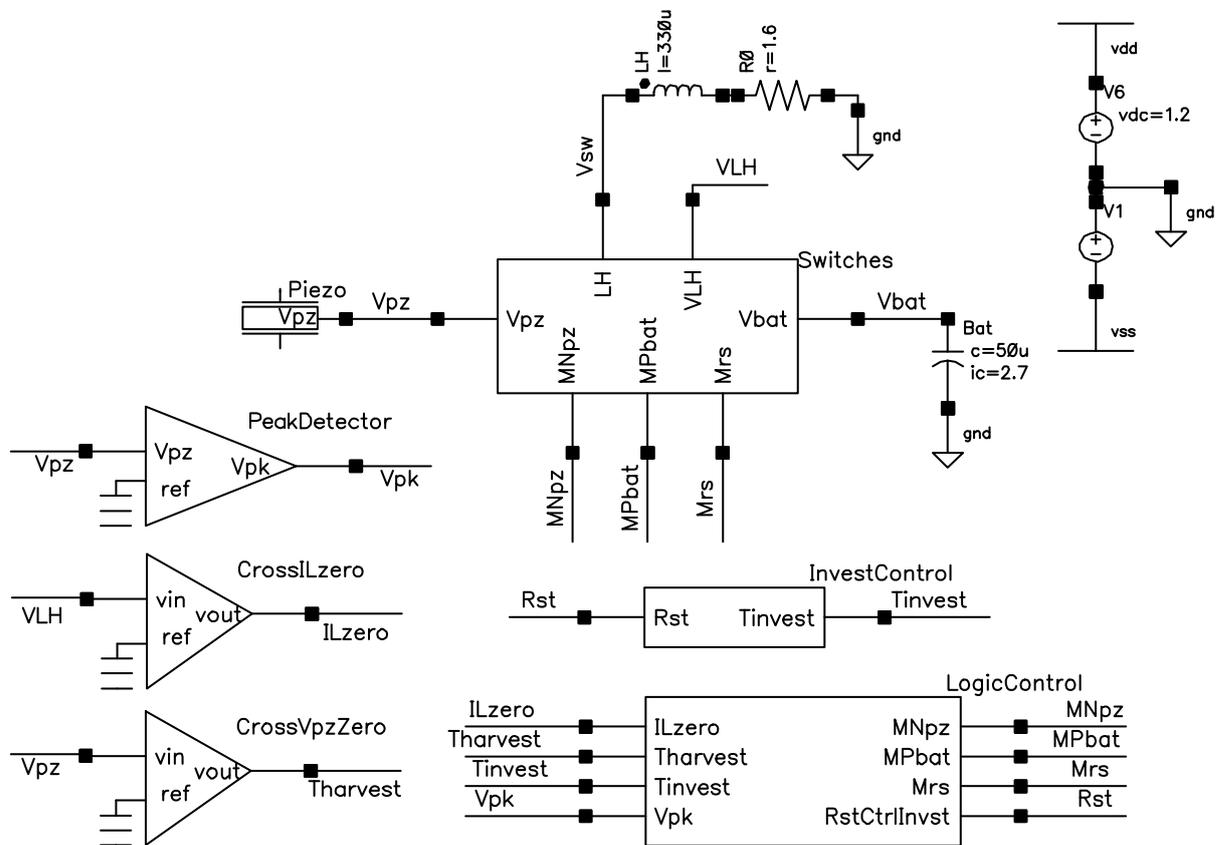
| ID ▲▲ | Title ▲▲ | Posted by ▲▲ | Actions ▲▲ |
|-------|-------------------------------------|------------------|------------|
| 4 | GF130 TUTORIAL | ANDRE AITA | |
| 7 | COMO COMPILAR PROJETOS NO IC-COLLAB | PETER PREVEDELLO | |

Showing 1 to 2 of 2 entries Previous 1 Next

Fonte: Próprio Autor.

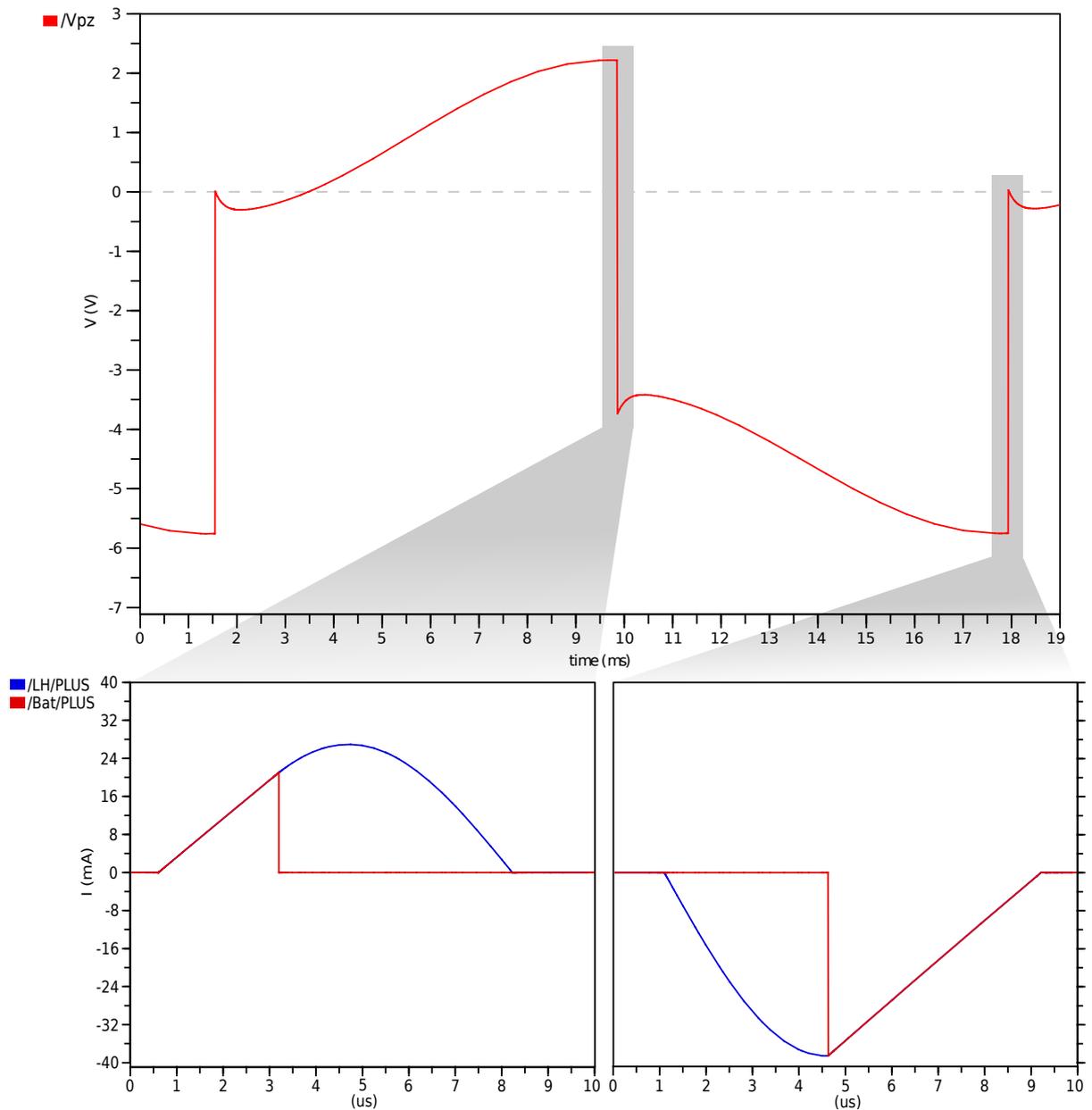
4.8 apresenta o esquemático do PiezoEnergy, após a compilação no repositório principal. A simulação é apresentada na Figura 4.9.

Figura 4.8 – Esquemático do Piezoelétrico após a compilação evidenciando os diversos blocos que modelam o sistema.



Fonte: Próprio Autor.

Figura 4.9 – Simulação elétrica do Piezoelétrico evidenciando a tensão V_{pz} sobre o piezo e as correntes no indutor LH e na bateria



Fonte: Próprio Autor.

5 CONCLUSÃO

Atualmente, observa-se que as universidades vem requerendo uma maior produtividade e eficiência durante o desenvolvimento de projeto de pesquisas (MORÁN, 2015).

Com o objetivo de dinamizar o trabalho realizado pelos professores e alunos do GMICRO da UFSM, observou-se a necessidade de desenvolver um ambiente virtual colaborativo para apoiar o ensino do projeto de CIs baseado em equipe, denominado neste trabalho como IC-COLLAB. Ao longo do presente trabalho, foram abordados diversos assuntos referente ao trabalho colaborativo e, como os ambientes virtuais podem torná-lo mais organizado e eficiente. Da mesma forma, estudos sobre o Modelo 3C de Colaboração foram essenciais para um melhor entendimento sobre como a colaboração ocorre através da inter-relação das dimensões comunicação, coordenação e cooperação.

Para o desenvolvimento e implementação do ambiente virtual IC-COLLAB, utilizou-se o Ciclo de Desenvolvimento de *Groupware*, que combina os métodos espiral e sequencial da engenharia de *software*. O ambiente foi planejado para lidar com as questões frequentemente observadas durante o projeto de tais circuitos em cursos acadêmicos, como a falta de colaboração entre participantes da equipe (ou dificuldades para colaborar), falta de organização, comunicação e documentação. O Ciclo de Desenvolvimento de *Groupware* é dividido em fases e primeiramente o desenvolvedor partiu da análise do domínio e com base nos conceitos do Modelo 3C foram classificados os componentes necessários para oferecer suporte computacional à dinâmica estabelecida para a colaboração. Do mesmo modo, o levantamento de requisitos foi realizado com base na análise baseada no Modelo 3C. Para a modelagem e documentação do projeto utilizou-se a linguagem UML, a qual facilita o entendimento de algumas especificações levantadas na análise de requisitos. Na fase de implementação, toda a análise e documentação realizada anteriormente foi traduzida para uma linguagem de programação no qual implementou-se um protótipo do ambiente virtual IC-COLLAB, resultando em uma primeira versão para a realização de testes e validação.

Desenvolvido sobre a plataforma *web*, o IC-COLLAB oferece todos os recursos essenciais para uma colaboração básica entre membros de uma equipe durante os projetos de CI, possibilitando o trabalho simultâneo em um ou mais projetos através do repositório distribuído, maior contato entre os projetistas através dos componentes de comunicação, bem como promove um processo decisório mais interativo e fluxo de trabalho mais organizado e eficiente.

A avaliação do ambiente IC-COLLAB, baseada em dois estudos de caso (Oscilador em Anel e Coletor de Energia Piezoelétrico) revelou resultados preliminares (observados e acordados entre os participantes de ambos os estudos de caso). É de conhecimento do autor deste trabalho que outros métodos de avaliação devem ser realizados, porém na houve tempo hábil para tal em função do processo de concepção, desenvolvimento

e implantação do IC-COLLAB. Abaixo são listados os resultados observados durante os estudos de caso.

- A organização global do projeto é muito melhorada, pois a abordagem de ensino colaborativo requer um trabalho coordenado. O aluno, ao projetar um circuito, têm em mente que seu circuito faz parte de um todo e estará sempre comprometido com suas tarefas;
- A segurança da informação do projeto é reforçada. O ambiente colaborativo IC-COLLAB, conforme descrito e implementado aqui, fornece repositórios principais, locais e de versão que tornam o trabalho mais organizado e confiável. Considerando que o propósito principal é apoiar a colaboração, serve também como um sistema de *backup* e controle de versões para os arquivos dos circuitos submetidos para a compilação do projeto;
- Maior qualidade da documentação. No ambiente virtual IC-COLLAB, a documentação precisa ser entregue regularmente conforme as tarefas definidas pelo coordenador. Se o coordenador do projeto perceber que a documentação é fraca ou falha, ele/ela pode enviar um alerta para o aluno. Assim, no momento da compilação, a maior parte da documentação do projeto estará disponível;
- A colaboração em tempo real e a comunicação imediata entre os participantes estão disponíveis no ambiente como uma premissa para um ambiente de colaboração virtual, possibilitando a realização de reuniões não presenciais sobre o projeto;
- O compartilhamento das bibliotecas após a compilação final do projeto, armazenadas em pasta determinada, disponibilizando-as para serem utilizadas em futuros novos projetos.

Desta forma, conclui-se que o ambiente virtual IC-COLLAB revelou-se uma ferramenta eficiente para o apoio ao ensino de projetos de CIs desenvolvidos em equipe. Além de uma melhor organização dos projetos e oferecer meios de comunicação para cooperar e compartilhar conhecimento, o IC-COLLAB proporciona através do seu fluxo de trabalho, que o ensino e aprendizagem colaborativo aprimore não apenas o conhecimento de conteúdos técnicos e práticos, mas também a motivação, responsabilidade, tomada de decisão, trabalho em equipe e a liderança.

5.1 CONTRIBUIÇÕES

Com a utilização do ambiente virtual IC-COLLAB, podemos destacar algumas contribuições relevantes quando aplicado na prática durante o projeto de CIs em equipe:

- ambiente virtual desenvolvido sobre a plataforma *web* e com *layout* responsivo e interface com baixa carga cognitiva tornam a realização das atividades mais eficientes, pois os usuários não precisam aprender a lidar com a interface do ambiente;
- espera-se que uma decisão em equipe seja tecnicamente superior àquela decisão individual, pois o grupo possui mais informações e conhecimento, o que diminui a probabilidade de erros durante o projeto;
- maior produtividade durante as atividades realizadas, onde a equipe pode tornar o projeto mais qualificado na implementação dos resultados quando todos participam das tomadas de decisões, aumentando a compreensão do trabalho proposto e consequentemente obtendo resultados mais eficientes;
- o papel de coordenador dentro de cada projeto, encarregado de gerenciar todas as atividades da equipe. Sua função está em incluir ou remover participantes, criar bibliotecas, definir tarefas, acompanhar o andamento do trabalho e realizar a compilação das bibliotecas para o repositório principal. Diferente dos demais participantes, o coordenador tem acesso livre a todas as bibliotecas do repositório principal após a compilação do projeto;
- permite o trabalho colaborativo mesmo quando os participantes estão distantes geograficamente;
- a comunicação através dos componentes *chat* e/ou *e-mail* possibilita a realização de reuniões e consequentemente a troca de informações entre participantes distantes geograficamente.

5.2 TRABALHOS FUTUROS E PERSPECTIVAS

Ao término deste trabalho percebeu-se algumas necessidades e possibilidades de melhoria no ambiente virtual IC-COLLAB. E que, no entanto, devido à amplitude do trabalho e restrição de tempo, não puderam ser desenvolvidas, ficando as seguintes oportunidades de pesquisa:

- realizar um estudo de caso mais amplo, utilizando o ambiente IC-COLLAB em uma turma com um número de alunos a partir de dez;
- desenvolver e implantar mais utilidades na tela inicial do ambiente IC-COLLAB com informações mais detalhadas como: lista de usuários que estão acessando uma determinada aplicação no servidor; lista de usuários que estão utilizando mais memória do servidor; e/ou lista de usuários que estão consumindo mais processamento do servidor;

- desenvolver um assistente para realizar a instalação do ambiente virtual IC-COLLAB. O assistente tem como objetivo identificar os requisitos necessários para o ambiente e informar através de mensagens na tela se eles existem ou se é necessária a sua instalação. Por fim, o assistente auxilia o usuário a configurar a conexão com o banco de dados e definir a senha do administrador do ambiente;
- analisar e incluir os requisitos de segurança em sistemas *web* durante as fases do Ciclo de Desenvolvimento de *Groupware*. Segundo Sêmola (2013) e Mueller (2016), os sistemas *web* normalmente se tornam vulneráveis a ataques porque os desenvolvedores não tratam a segurança durante o processo de desenvolvimento ou não possuem conhecimento para tal;
- customizar o ambiente IC-COLLAB tornando-o flexível e configurável de modo que os usuários possam escolher quais os componentes são necessários para um determinado projeto;
- como a hierarquia de repositórios segue um modelo distribuído, o controle de versões é um ponto crítico e necessita ser melhorado quanto seu uso de espaço de armazenamento em disco. Desta forma, uma alternativa de melhoria seria desenvolver um componente que possibilite a integrar os sistemas GIT ou *SubVersion*, pois são sistemas específicos para o controle de versão.
- analisar o uso do ambiente virtual IC-COLLAB com outros ambientes de CAD, como a *Synopsys* e *Mentor Graphics*.

REFERÊNCIAS BIBLIOGRÁFICAS

- Apache. **What is the Apache HTTP Server Project?** 2017. Acesso em 10 nov. 2017. Disponível em: <https://httpd.apache.org/ABOUT_APACHE.html>.
- BARROS, L. A. Suporte a ambientes distribuídos para aprendizagem cooperativa. **Rio de Janeiro: UFRJ**, 1994.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. [S.l.]: Elsevier Brasil, 2006.
- BOOTSTRAP. **History**:: A brief overview of the history of bootstrap. 2017. Acesso em 09 nov. 2017. Disponível em: <<http://getbootstrap.com/docs/4.0/about/history>>.
- BOTELHO, E. X.; VIDAL, J. Cscw-trabalho cooperativo suportado por computador. **HOLLOS**, Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, v. 1, 2005.
- BRITO, R. F. D.; PEREIRA, A. T. C. Um estudo para ambientes colaborativos e suas ferramentas. In: **CONAHPA-Congresso Nacional de Ambientes Hipermídia para Aprendizagem**. [S.l.: s.n.], 2004.
- BRNA, P. Modelos de colaboração. **Revista Brasileira de Informática na Educação**, v. 3, n. 1, p. 9–16, 1998.
- BULGRAEN, V. C. O papel do professor e sua mediação nos processos de elaboração do conhecimento. **Revista conteúdo**, v. 1, n. 4, p. 30–38, 2010.
- CentOS. **CentOS Linux**. 2018. Acesso em 06 fev. 2018. Disponível em: <<https://www.centos.org/about/>>.
- CHACON, S.; STRAUB, B. **Distributed**. 2017. Acesso em 15 nov. 2017. Disponível em: <<https://git-scm.com/about/distributed>>.
- COUGO, P. **Modelagem conceitual e projeto de banco de dados**. [S.l.]: Elsevier Brasil, 2013.
- DEHAENE, W. et al. Circuits and systems engineering education through interdisciplinary team-based design projects. In: IEEE. **Circuits and Systems (ISCAS), 2011 IEEE International Symposium on**. [S.l.], 2011. p. 1195–1198.
- DEWAN, P. An integrated approach to designing and evaluating collaborative applications and infrastructures. **Computer Supported Cooperative Work (CSCW)**, Springer, v. 10, n. 1, p. 75–111, 2001.
- DILLENBOURG, P. **Collaborative learning: Cognitive and computational approaches. advances in learning and instruction series**. [S.l.]: ERIC, 1999.
- ELLIS, C. A.; GIBBS, S. J.; REIN, G. Groupware: some issues and experiences. **Communications of the ACM**, ACM, v. 34, n. 1, p. 39–58, 1991.
- FAGUNDES, L. Educação a distância em ciência e tecnologia: o projeto educadi/cnpq–1997. **Em Aberto**, v. 16, n. 20, 1996.

FERNANDES, J. R. An analog integrated systems course proposal for a 2nd cycle of studies of the bologna process. In: IEEE. **Circuits and Systems (ISCAS), 2016 IEEE International Symposium on**. [S.l.], 2016. p. 133–136.

FILHO, J. A. de C. et al. Portal humanas: Um ambiente colaborativo para criação de projetos e comunidades virtuais para a área de humanidades. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. [S.l.: s.n.], 2005. v. 1, n. 1, p. 254–264.

FILIPPO, D. et al. Ambientes colaborativos de realidade virtual e aumentada. **C. Kirner, R. Siscoutto, EDS**, p. 168–191, 2007.

FISCHER, G. et al. Seeding, evolutionary growth and reseeding: The incremental development of collaborative design environments. **Coordination theory and collaboration technology**, Citeseer, v. 447, p. 472, 2001.

FUKS, H.; GEROSA, M. A.; LUCENA, C. J. P. de. The development and application of distance learning courses on the internet. **Open Learning**, Taylor & Francis, v. 17, n. 1, p. 23–38, 2002.

FUKS, H.; PIMENTEL, M. **Sistemas colaborativos**. [S.l.]: Elsevier Brasil, 2011.

FUKS, H.; RAPOSO, A. B.; GEROSA, M. A. Engenharia de groupware: desenvolvimento de aplicações colaborativas. In: **XXI Jornada de Atualização em Informática, Anais do XXII Congresso da Sociedade Brasileira de Computação**. [S.l.: s.n.], 2002. v. 2, p. 89–128.

FUKS, H. et al. Do modelo de colaboração 3c à engenharia de groupware. **Simpósio Brasileiro de Sistemas Multimídia e Web-Webmídia**, p. 0–8, 2003.

GEROSA, M. A. Desenvolvimento de groupware componentizado com base no modelo 3c de colaboração. **Universidade Católica do Rio de Janeiro (PUC-Rio)**, 2006.

GILAB. **About US**. 2017. Acesso em 23 mar. 2018. Disponível em: <<https://about.gitlab.com/about/>>.

GROSZ, B. J. Collaborative systems (aaai-94 presidential address). **AI magazine**, v. 17, n. 2, p. 67, 1996.

GUTWIN, C.; GREENBERG, S. The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces. In: IEEE. **Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000.(WET ICE 2000). Proceedings. IEEE 9th International Workshops on**. [S.l.], 2000. p. 98–103.

HARRISON, D. S. et al. Electronic cad frameworks. **Proceedings of the IEEE**, IEEE, v. 78, n. 2, p. 393–417, 1990.

HEUSER, C. A. **Projeto de banco de dados: Volume 4 da Série Livros didáticos informática UFRGS**. [S.l.]: Bookman Editora, 2009.

HIRAMA, K. **Engenharia de software: qualidade e produtividade com tecnologia**. [S.l.]: Elsevier Brasil, 2011.

HTML. **Introdução ao HTML**. 2017. Acesso em 09 nov. 2017. Disponível em: <<https://developer.mozilla.org/pt-BR/docs/HTML/Introduction>>.

IONESCU, B. et al. A chat-centric collaborative environment for web-based real-time collaboration. In: IEEE. **Applied Computational Intelligence and Informatics (SACI), 2015 IEEE 10th Jubilee International Symposium on**. [S.l.], 2015. p. 105–110.

JR, H. E. **Engenharia de Software na prática**. [S.l.]: Novatec Editora, 2010.

KERBER, A. et al. Assessing device reliability margin in scaled cmos technologies using ring oscillator circuits. In: **2017 IEEE Electron Devices Technology and Manufacturing Conference (EDTM)**. [S.l.: s.n.], 2017. p. 28–30.

KWON, D.; RINCÓN-MORA, G. A. A single-inductor 0.35 μm cmos energy-investing piezoelectric harvester. **IEEE Journal of Solid-State Circuits**, v. 49, n. 10, p. 2277–2291, Oct 2014. ISSN 0018-9200.

LAURILLAU, Y.; NIGAY, L. Clover architecture for groupware. In: ACM. **Proceedings of the 2002 ACM conference on Computer supported cooperative work**. [S.l.], 2002. p. 236–245.

LEITE, H. P.; BONOMO, I. d. S. Análise comparativa de projeto e administração de banco de dados entre os sgbd's cassandra e mysql. 2017.

LUKOSCH, S.; SCHÜMMER, T. Groupware development support with technology patterns. **International Journal of Human-Computer Studies**, Elsevier, v. 64, n. 7, p. 599–610, 2006.

MACMILLEN, D. et al. An industrial view of electronic design automation. **IEEE transactions on computer-aided design of integrated circuits and systems**, IEEE, v. 19, n. 12, p. 1428–1448, 2000.

MALUCELLI, A. et al. Sistema de informação para apoio à sistematização da assistência de enfermagem. **Revista Brasileira de Enfermagem**, Associação Brasileira de Enfermagem, v. 63, n. 4, p. 629–636, 2010.

MALVINO, A. P.; BATES, D. J. **Eletrônica-Vol. 2: 8ª Edição**. [S.l.]: McGraw Hill Brasil, 2016.

MARCELO, A. **Apache: configurando o servidor WEB para Linux**. [S.l.]: Brasport, 2005.

MARÓTI, M. et al. Online collaborative environment for designing complex computational systems. **Procedia Computer Science**, Elsevier, v. 29, p. 2432–2441, 2014.

MILANI, A. **Construindo aplicações web com PHP e MySQL**. São Paulo: Novatec Editora, 2010. 529 p.

Molund, Stefan. **Cadence**. 2017. Acesso em 13 nov. 2017. Disponível em: <<http://www.eit.lth.se/cadsys/cadence.html>>.

MORÁN, J. Mudando a educação com metodologias ativas. **Coleção Mídias Contemporâneas. Convergências Midiáticas, Educação e Cidadania: aproximações jovens**, v. 2, p. 15–33, 2015.

Morandini, Marcelo. **Crterios e Requisitos para Avaliação da Usabilidade de Interfaces em Groupware - CSCW**. 2017. Acesso em 09 nov. 2017. Disponível em: <<http://www.dca.fee.unicamp.br/courses/IA368F/1s1998/Monografias/morandini.html>>.

MUELLER, J. **Segurança para desenvolvedores web: Usando JavaScript, HTML e CSS**. [S.l.]: NOVATEC, 2016. ISBN 9788575224847.

PEINADO, J.; GRAEML, A. R. **Administração da produção**. Curitiba: UnicenP, 2007. 750 p.

PHP. **O que é o PHP?** 2017. Acesso em 10 nov. 2017. Disponível em: <https://secure.php.net/manual/pt_BR/intro-what-is.php>.

PIMENTEL, M.; FUKS, H.; LUCENA, C. J. Um processo de desenvolvimento de sistemas colaborativos baseado no modelo 3c: Rup-3c-groupware. **Anais do IV SBSI. SBSI**, v. 7, 2008.

PIMENTEL, M. et al. Modelo 3c de colaboração para o desenvolvimento de sistemas colaborativos. **Anais do III Simpósio Brasileiro de Sistemas Colaborativos**, p. 58–67, 2006.

PMI. **A Guide to the Project Management Body of Knowledge (PMBOK Guide), 4th Edition**. [S.l.]: Project Management Institute, 2008.

PRESMAN, R. S. **Engenharia de Software: Uma abordagem Profissional**. Porto Alegre: AMGH, 2011. 780 p.

QI, C.-X. et al. Web-based collaborative learning system. In: IEEE. **Networking and Digital Society (ICNDS), 2010 2nd International Conference on**. [S.l.], 2010. v. 2, p. 557–560.

RAPOSO, A. B. et al. Prescribing e-learning activities using workflow technologies. In: CI-TESEER. **Computer Supported Activity Coordination**. [S.l.], 2004. p. 71–80.

ROSA, J. de la. A teamwork-based education strategy for teaching lab of analog integrated circuits design. In: IEEE. **Global Engineering Education Conference (EDUCON), 2012 IEEE**. [S.l.], 2012. p. 1–4.

SAWICKI, S.; INDRUSIAK, L. S.; REIS, R. A. d. L. Projeto cooperativo no ambiente cave. In: **WORKSHOP IBERCHIP**. [S.l.: s.n.], 2002. v. 8.

SCHAF, F. M. Arquitetura modular para ambientes virtuais de ensino de automação com suporte à realidade mista e colaboração. 2011.

SCHLICHTER, J.; BORGHOFF, U. **Computer-Supported Cooperative Work: Introduction to Distributed Applications**. [S.l.]: Springer Berlin, 2000.

SNOWDON, D.; CHURCHILL, E. F.; MUNRO, A. J. Collaborative virtual environments: Digital spaces and places for cscw: An introduction. In: SPRINGER. **Collaborative virtual environments**. [S.l.], 2001. p. 3–17.

SOMMERVILLE, I. **Engenharia de Software**. São Paulo: Pearson Prentice Hall, 2011. 529 p.

SONEGO, A. H. S. et al. Use of moodle as a tool for collaborative learning: a study focused on wiki. **IEEE Revista Iberoamericana de Tecnologias del Aprendizaje**, IEEE, v. 9, n. 1, p. 17–21, 2014.

Subversion. **Apache Subversion**. 2018. Acesso em 24 abr. 2018. Disponível em: <<https://subversion.apache.org/>>.

SÊMOLA, M. **Gestão da Segurança da Informação: Uma Visão Executiva**. [S.l.]: Elsevier, 2013. ISBN 9788535271782.

WOLF, A. S.; SILVA, M. S. D. Ferramenta para gerenciamento e apoio ao desenvolvimento distribuído de software. **Revista Destaques Acadêmicos**, v. 5, n. 4, 2013.

ANEXO A – REQUISITOS FUNCIONAIS (RF)

RF 01: **Cadastrar Usuários**

Descrição: Usuário administrador cadastra, altera e exclui usuários. Deverá existir também uma opção para desabilitar o usuário;

Caso de Uso: Gerenciar Usuários;

RF 02: **Criar base de dados centralizada para o usuário**

Descrição: O sistema deverá criar um diretório com o nome do login do usuário cadastrado. Dentro deste diretório, o sistema deverá criar outro com o nome VERSAO, local onde será realizado o controle de versões dos circuitos do usuário;

Caso de Uso: Criar base de dados do usuário;

RF 03: **Definir perfil do Usuário**

Descrição: Usuário administrador deve definir se o perfil do Usuário será de Administrador ou de Participante.;

Caso de Uso: Definir perfil do Usuário;

RF 04: **Solicitar autenticação**

Descrição: Todos os usuários devem autenticar-se para utilizar o sistema.;

Caso de Uso: Solicitar autenticação;

RF 05: **Cadastrar modelo padrão do arquivo cds.lib**

Descrição: Usuário administrador cadastra, altera e exclui informações no modelo padrão do arquivo cds.lib;

Caso de Uso: Cadastrar modelo padrão do arquivo cds.lib;

RF 06: **Cadastro de Projetos**

Descrição: Usuário administrador cadastra, altera e exclui projetos no sistema;

Caso de Uso: Cadastro de Projetos.

RF 07: **Criar base de dados centralizada para o projeto**

Descrição: O sistema deverá criar um diretório com o nome do título do projeto cadastrado (repositório central). O sistema também deverá criar esse mesmo diretório na pasta do usuário (repositório local) quando ele for adicionado ao projeto;

Caso de Uso: Criar base de dados centralizada para o projeto;

RF 08: Acessar diretórios e arquivos do repositório local

Descrição: O sistema deve permitir que os usuários autenticados no sistema tenham permissões de leitura e escrita nos diretórios e arquivos dos seus repositórios locais;

Caso de Uso: Acessar diretórios e arquivos do repositório local;

RF 09: Adicionar coordenador no projeto

Descrição: O usuário administrador deverá adicionar pelo menos um participante e defini-lo como coordenador;

Caso de Uso: Adicionar coordenador no projeto;

RF 10: Acessar diretórios e arquivos do repositório central

Descrição: O sistema deve permitir que o coordenador do projeto tenha permissões de leitura e escrita nos diretórios e arquivos do repositório central;

Caso de Uso: Acessar diretórios e arquivos do repositório central;

RF 11: Gerenciar Projetos

Descrição: O coordenador adiciona e remove novos participantes (usuários) ao projeto. O coordenador também cadastra, altera e exclui Circuitos e Tarefas para outros participantes;

Caso de Uso: Gerenciar Projetos;

RF 12: Adicionar participante no projeto

Descrição: O coordenador adiciona e remove participantes no projeto;

Caso de Uso: Adicionar participante no projeto;

RF 13: Gerenciar repositório do projeto

Descrição: O coordenador pode visualizar as pastas e arquivos do projeto;

Caso de Uso: Gerenciar repositório do projeto;

RF 14: Cadastrar Biblioteca

Descrição: O coordenador cadastra bibliotecas para os participantes;

Caso de Uso: Cadastrar Biblioteca;

RF 15: Gerenciar Bibliotecas

Descrição: Participante do projeto adiciona, altera e exclui dados no repositório da biblioteca ao qual está trabalhando. O participante também deverá registrar as alterações quando estas ocorrerem;

Caso de Uso: Gerenciar Biblioteca;

RF 16: Submeter alterações das Bibliotecas

Descrição: O participante poderá submeter alterações da biblioteca;

Caso de Uso: Submeter alterações das Bibliotecas;

RF 17: Gerenciar alterações das Bibliotecas

Descrição: O sistema deverá realizar o gerenciamento de alterações das bibliotecas submetidas. Esse gerenciamento deverá ser realizado através do controle de versões, onde a cada nova submissão realizada pelo participante, uma nova versão é gerada;

Caso de Uso: Gerenciar alterações das Bibliotecas;

RF 18: Restaurar Bibliotecas

Descrição: O participante poderá restaurar uma versão antiga da sua biblioteca;

Caso de Uso: Submeter alterações das Bibliotecas;

RF 19: Cadastrar Tarefas

Descrição: O coordenador cadastra Tarefas para os participantes;

Caso de Uso: Cadastrar Tarefas;

RF 20: Gerenciar Tarefas

Descrição: Participante do projeto adiciona tarefas e altera acompanhamentos na sua tarefa;

Caso de Uso: Gerenciar Tarefas;

RF 21: Gerenciar documentos

Descrição: Usuários poderão criar, editar e compartilhar documentos de ajuda no sistema.;

Caso de Uso: Gerenciar documentos;

RF 22: Compilar Bibliotecas do Projeto

Descrição: Somente o Coordenador poderá compilar o conjunto de bibliotecas para o repositório central do projeto;

Caso de Uso: Compilar Bibliotecas do Projeto;

RF 23: Comunicação síncrona

Descrição: Participantes poderão se comunicar com outros participantes instantaneamente;

Caso de Uso: Comunicação síncrona;

RF 24: Comunicação assíncrona

Descrição: Participantes poderão deixar mensagens para outros participantes acessarem mais tarde;

Caso de Uso: Comunicação assíncrona;

RF 25: Gerenciar Fórum

Descrição: Participantes poderão criar e responder listas de discussão;

Caso de Uso: Gerenciar Fórum;

RF 26: Gerenciar arquivo cds.lib

Descrição: Somente o coordenador de projeto poderá realizar alterações no arquivo cds.lib do projeto;

Caso de Uso: Gerenciar arquivo cds.lib;

ANEXO B – REQUISITOS NÃO-FUNCIONAIS (RNF)

RNF 01: **O sistema deve ser *web***

Descrição: O sistema deverá ser *web*;

RNF 02: **Desenvolvido em PHP**

Descrição: O sistema deverá ser desenvolvido utilizando a linguagem de programação PHP;

RNF 03: **Deve rodar nos navegadores Firefox e Chrome**

Descrição: O sistema deverá rodar nos navegadores Mozilla Firefox e Google Chrome;

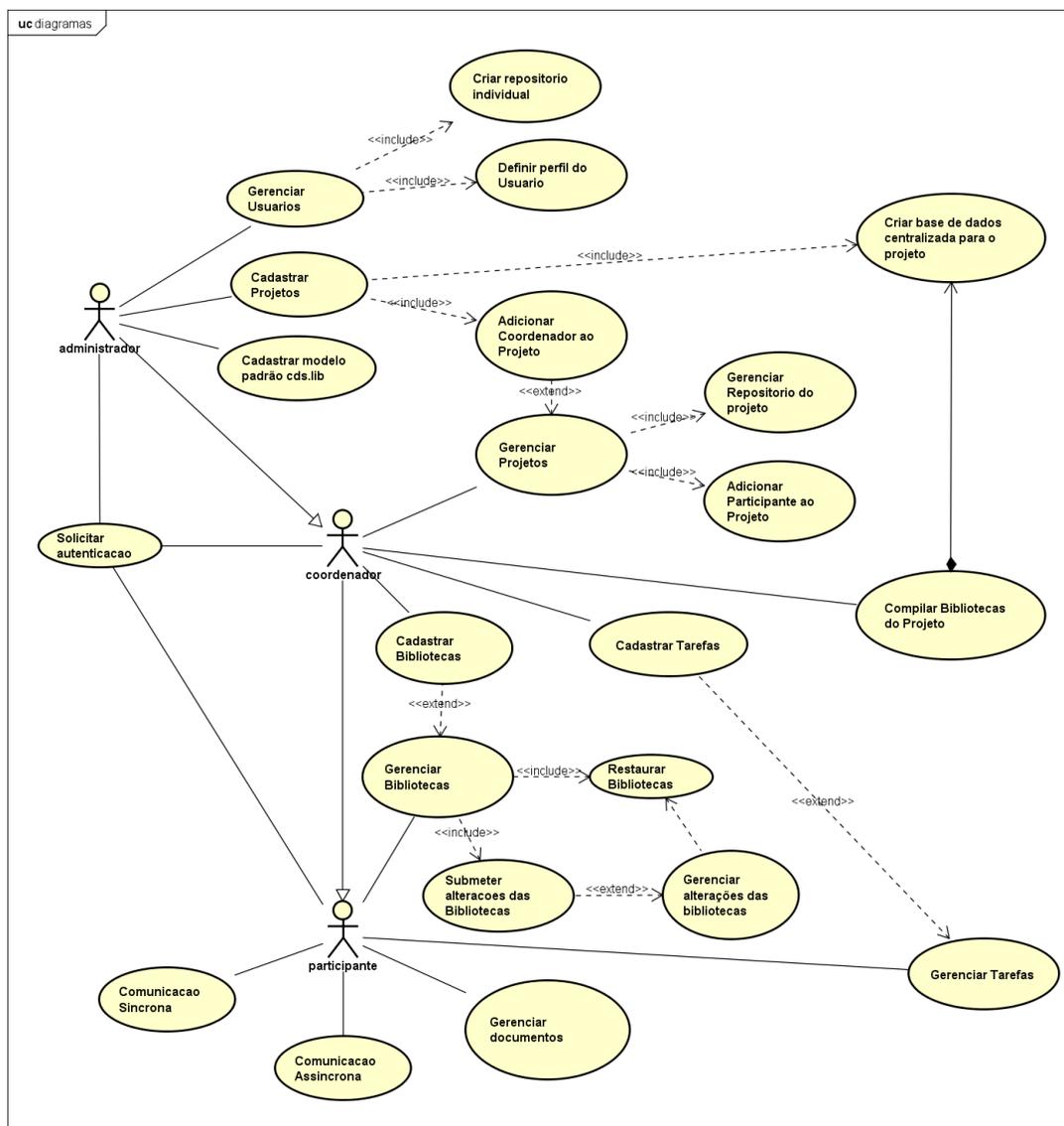
RNF 04: **Disponibilizar informações de desempenho do servidor**

Descrição: O sistema deverá disponibilizar as seguintes informações de desempenho do servidor:

- uso do processador;
- uso da memória RAM;
- espaço em disco disponível;
- nome do sistema operacional;
- nome da distribuição do sistema operacional;
- versão da distribuição do sistema operacional;
- nome dos usuários que estão conectados remotamente no servidor.

ANEXO C – DIAGRAMA DE CASO DE USO

Figura C.1 – Diagrama de Casos de Uso do IC-COLLAB.

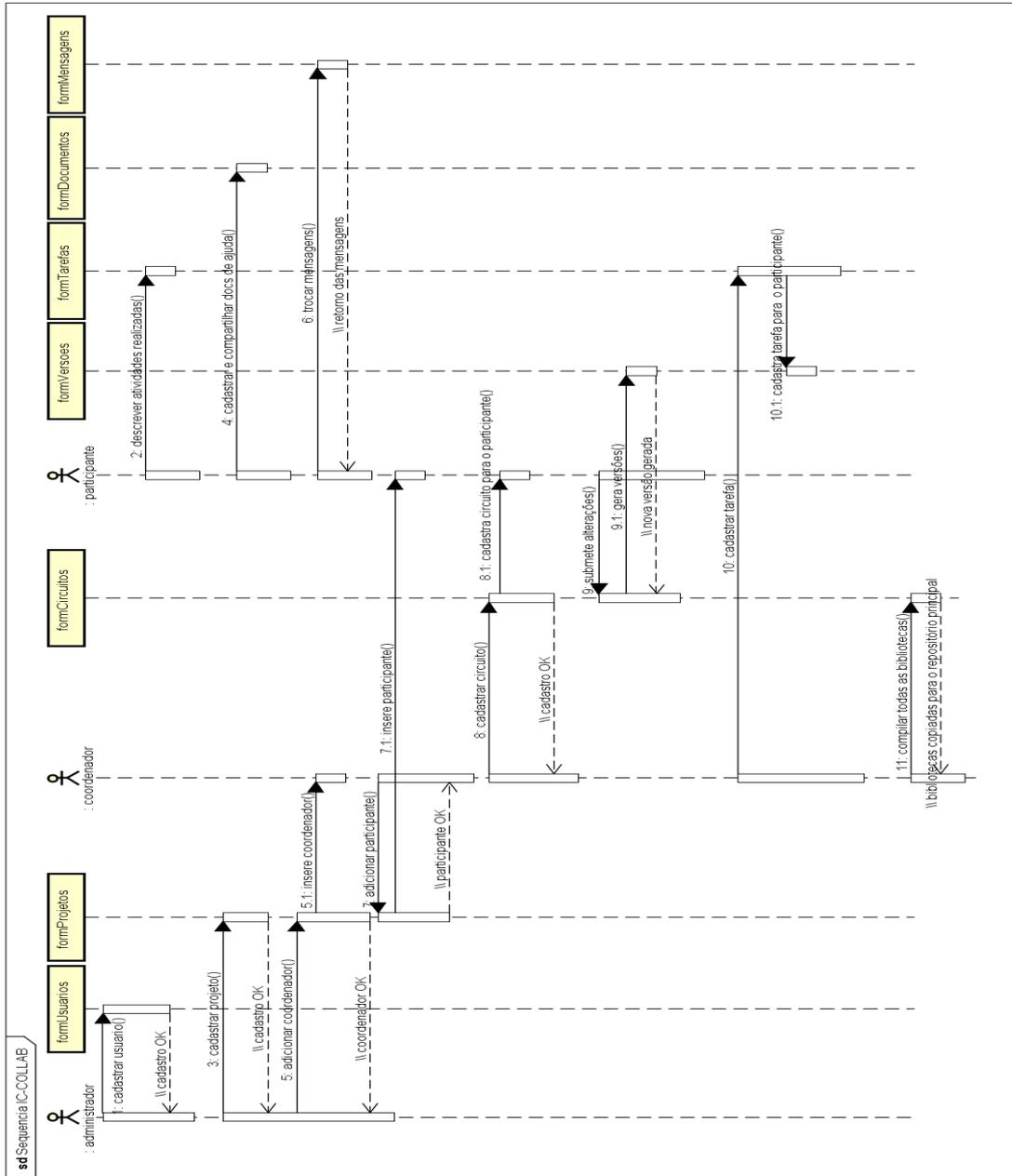


powered by Astah

Fonte: Próprio autor.

ANEXO D – DIAGRAMA DE SEQUÊNCIA

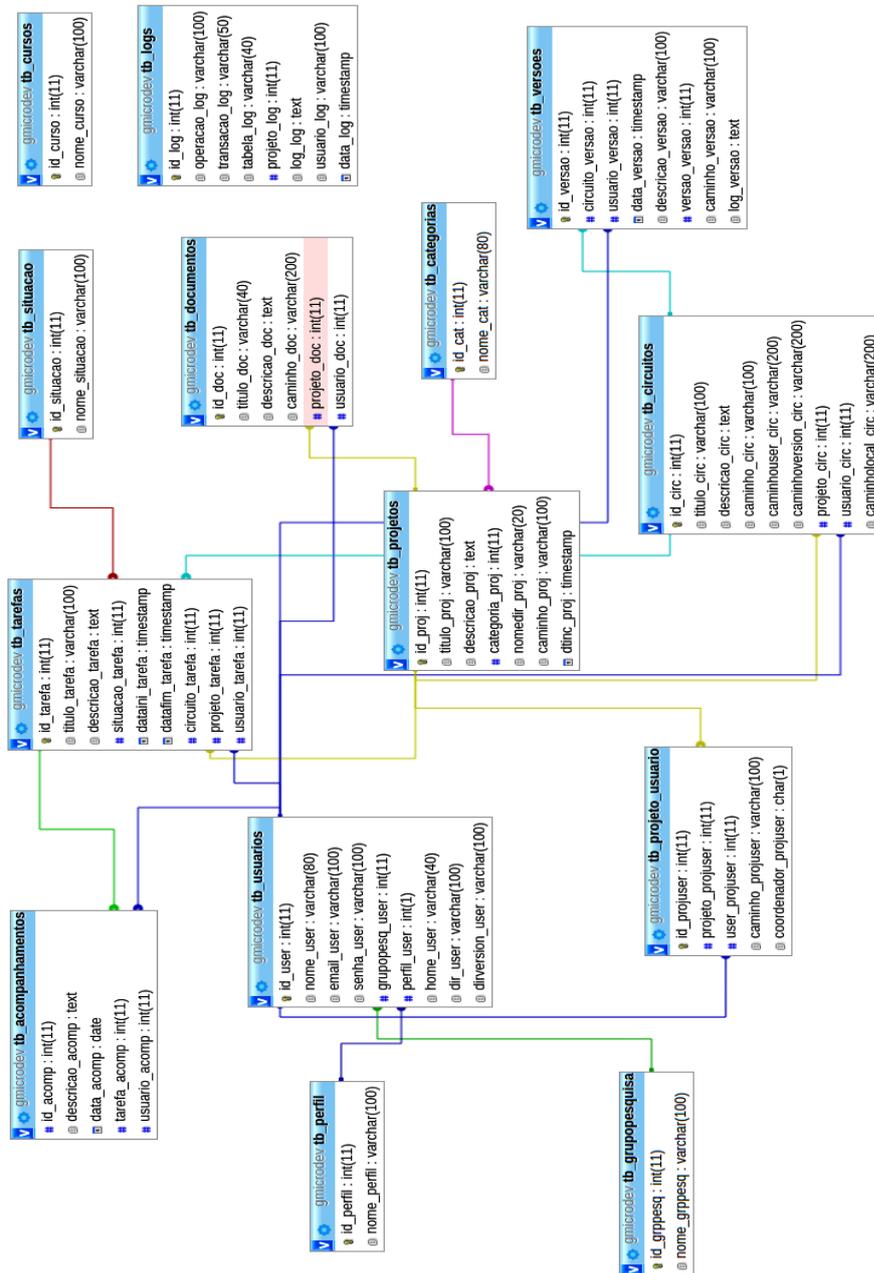
Figura D.1 – Diagrama de Sequência do IC-COLLAB.



Fonte: Próprio autor.

ANEXO E – DIAGRAMA ENTIDADE-RELACIONAMENTO

Figura E.1 – Diagrama Entidade-Relacionamento do IC-COLLAB.



Fonte: Próprio autor.

ANEXO F – ARTIGO PUBLICADO NO ISCAS 2018

Artigo intitulado "*Virtual Collaborative Environment for Support of Teamwork-Based Education of Analog IC Design*" foi publicado no ano de 2018 no *International Symposium on Circuits and Systems (ISCAS)*, principal conferência da *IEEE Circuits and Systems (CAS) Society*.

Virtual Collaborative Environment for Support of Teamwork-based Education of Analog IC Design

Peter Prevedello

Post-Graduation Course in Computer Science - PGCC
Federal University of Santa Maria, UFSM
Santa Maria, RS, Brazil 97105-900
peterprevedello@gmail.com

Andre L. Aita

Department of Electronics and Computing
Federal University of Santa Maria, UFSM
Santa Maria, RS, Brazil 97105-900
aaita@inf.ufsm.br

Abstract—This work presents a virtual collaborative environment for support of teamwork-based education of Integrated Circuit (IC) design. The environment was planned to cope with those issues often observed during the design of such circuits in academic courses as lack of collaboration among students or difficulties to collaborate to, lack of organization and documentation. By providing essential means for collaboration, a case study has revealed these issues are diminished and a clean straight-forward design flow for teamwork-based education is achievable.

Keywords— *Teamwork-based Project-based Education, Collaborative Environment, 3C Collaboration Model, Analog IC Design.*

I. INTRODUCTION

Remarkable advances in microelectronics have allowed the design of increasingly complex integrated circuits (ICs) and systems. Helped by Computer Aided Design (CAD), e.g. IC Design Frameworks, and world-wide available information and communication technologies (ICTs), the building of complex integrated systems is undoubtedly a reality. Also is the fact that, in industry, most of the complex ICs and integrated systems are designed following a team-based approach.

In the academia, while not a rule, team-based approach is also reality. For instance, in [1] the authors report their conviction that the most appropriate way of teaching engineering is through project-based and learning by experience, carried out by groups of students.

A similar approach is reported in [2] where a teamwork-based methodology to teach in analog IC design courses is proposed. According to the text, the students are encouraged to be an active part of the course working together towards the common objective of designing some case-study circuits.

Also in [3], the author suggests a pedagogical model chosen to emphasize the practical design of an IC complying with a set of specifications. Whereas the text focuses on the program content of an IC design course, the teaching method follows a project-based approach which is inline with the previously mentioned works.

However, before accounting the advantages of a project- and teamwork-based teaching approach, an environment to assist course-enrolled students, provide organization and cooperation means for right work execution, is essential.

And despite of the wide use of collaborative systems for software development and computational systems [4], despite

of all design resources provided by IC Design Frameworks and other assistive ICTs, specific needs for support of teamwork-based education of analog IC design remain unaddressed.

With this in mind, this work presents a virtual collaborative environment for support of academic teaching of analog IC design, focusing on providing minimum means to cope with the needs of project- and team-based models. The virtual environment provides tools for collaboration, guaranteeing project organization and efficient cooperation among participants, enhancing not only individual practices, but also teamwork and leadership. An organized main repository allows saving and sharing libraries individually developed by the students. Using a version controller, each member is able to track and manage changes in his/her libraries. Supporting materials and tutorials can be created and further shared with the participants.

Section II presents the fundamentals of collaborative work. Section III describes the virtual collaboration environment, its implementation, architecture and organization. A case study with results is presented in Section IV, followed by conclusions.

II. FUNDAMENTALS OF COLLABORATIVE ENVIRONMENTS

The main goal of a collaborative environment (or Groupware) is to manage and help a group of people towards the execution of a common task, e.g. development of an integrated system, with an IC Design Framework, by providing an organized and shared interface where all students act together effectively. Many of the required elements for the development of such Groupwares derive from software engineering. The collaboration model and Groupware development strategy are introduced here.

A. 3C Collaboration Model

The development of virtual collaborative environments or Groupwares requires a full and deep understanding of the collaboration concept. The 3C Collaboration Model, as originally proposed by [5], established three fundamental dimensions, for instance, Communication, Coordination and Cooperation, as presented in Fig. 1, to help analyzing, classifying and developing collaborative environments. According to the 3C

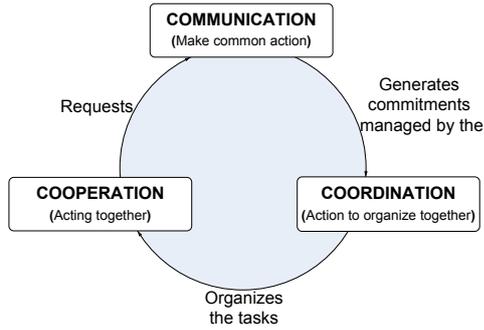


Fig. 1. 3C Collaboration Model or Chart (adapted from [6]).

Collaboration Model, in a collaborative environment, people have to be able to [6][7]:

- change information (communication);
- get themselves organized (coordination); and
- work together (cooperation) in a shared working space.

As the three dimensions are interdependent, a collaborative environment must support all of three, even though, if, given the application needs, one dimension is prioritized.

B. Groupware Development Cycle

Fig. 2 shows the Groupware Development Cycle, proposed by [6], which is based on the spiral model used for software developing. The main advantage of such model is the ability to combine the sequential model with the prototyping interactivity, which is essential for a groupware development.

According to Fig. 2, the Domain Analysis cycle is based on the 3C Collaboration Model. During this phase, prospective applications to support the communication, coordination and cooperation needs are studied.

Functional and non-functional communication, coordination and cooperation needs are further addressed in the Requirement Analysis Cycle. Often these needs are unclear, not allowing an accurate specification of the application.

In the Design Cycle, the processes defined during the requirement analysis need to be mapped and documented by means of tools and extensions like the Unified Modeling Language (UML).

The Collaborative Environment will be developed during the Implementation Cycle. This implementation is further made available for tests during the Test Cycle, aiming to detect issues, failures or deficiencies in the environment. The cycle is initiated again, until the system is considered ready for work.

III. IC VIRTUAL COLLABORATIVE ENVIRONMENT

A. System Development

The virtual collaborative environment was developed according to the Groupware Development Cycle. The programming language used in this development was PHP [8], a script language thought for the development of dynamic WWW applications as it runs on the server side and can be incorporated to HyperText Markup Language (HTML).

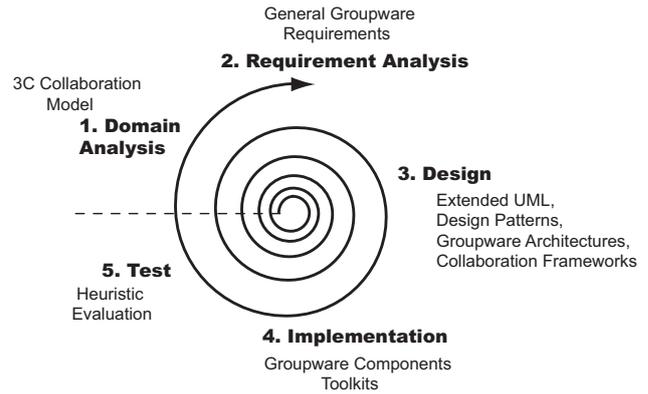


Fig. 2. Classical spiral model with typical software development cycles.

The layout environment was implemented using the Bootstrap Framework, for responsive layouts. To guarantee a reliable data storage, MySQL was adopted, which is a Data Bank Management System (DBMS) that uses the language SQL (Structured Query Language) as interface. All the system operations are triggered from a graphical interface.

B. System Architecture and Organization

Fig. 3 shows the collaborative system architecture, where the main and local repositories are depicted. The repositories management is based on the distributed model of GIT [9] (distributed because users 1 to N have their own local repositories) which allows the team-based projects 1 to M to be structured and compiled hierarchically in the main repository. With this architecture, the project structure is transparent for all users and collaboration and team-based designs eased. Fig. 3 also shows that a given user may participate in more that one project (i.e. the architecture is multi-project multi-user) without compromising organization.

A version controller with essential functions was developed to manage the libraries in the local repository, allowing each member to perform versioning (and backup) of his/her work. For an improved architecture, the library versions will be stored in a specific directory (e.g. VERSIONS) inside the local repository. It is worth noticing that the hierarchy created for a project inside the main repository is also created inside the local repository (under both USER and VERSIONS directories as shown in Fig. 3), as this replication is needed for further final project compilation.

Regarding the system organization, each project has one coordinator, who is responsible for tasks assignment. With a commit command, a student may respond to an assignment by uploading for instance a given library to his/her local repository. Additionally, a second (and further ones) commit command will copy the current library to the local version repository before uploading the newer library, so as to build the versioning. The commit command will thus in turn:

- generate library backups, adding safety to the project;
- allow users to later restore a former working library, with an update command;

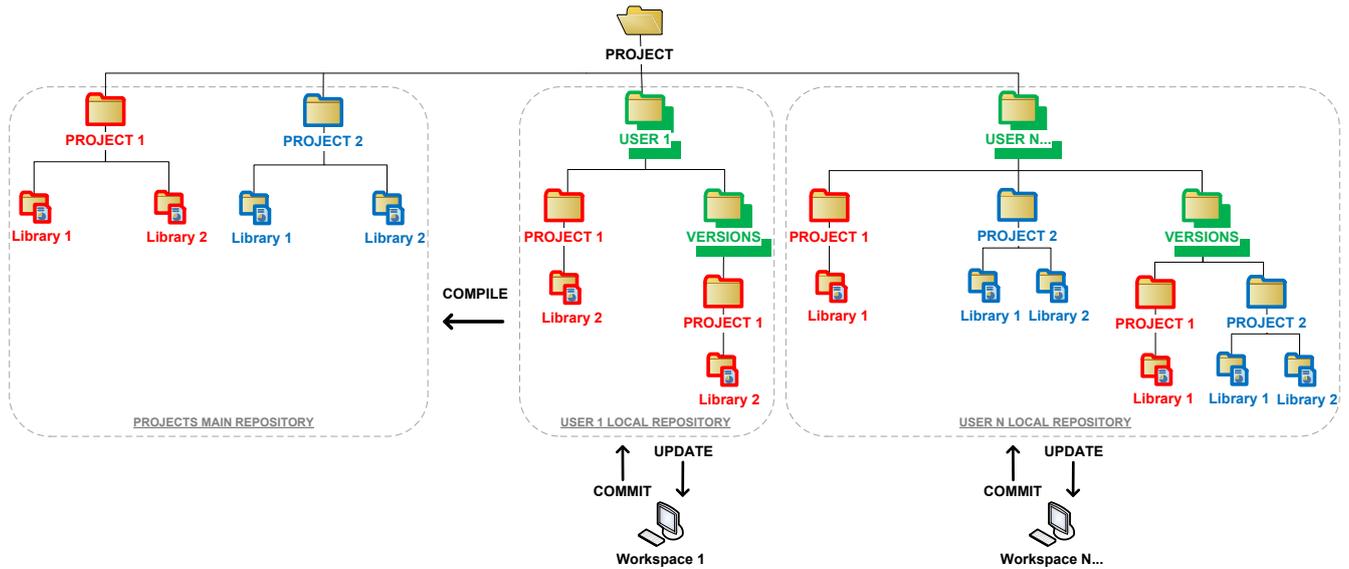


Fig. 3. Collaborative system architecture and organization.

- provide stable libraries files for compilation.

After all users have responded and committed their tasks, the coordinator may start a project compilation, copying the libraries from the local repositories and start building the full project in the main repository. Further compilations will only copy libraries that were modified in their local repositories.

IV. CASE STUDY AND RESULTS

In order to evaluate the developed collaborative system IC-Collab, a ring oscillator was proposed as a case study. First, the students were registered into the system. For each user registered, a user local repository was created (with folders USER and VERSIONS, as depicted in Fig. 3).

In sequence, the project was created, helped by a four-step procedure wizard. In the first step, the title was defined with attached description (as shown in Fig. 4). The project title, in this case study RINGOSCILLATOR, defines the main repository, and because of that, needs to be unique in the environment. In the following step, the students and all team members were added to the project. Among them, one was assigned as project coordinator¹ (third step), and was thus responsible by tasks proposals and management (including the final project compilation). The last step assigned, for each member, the libraries (or cells) each one would be responsible for (as shown in Fig. 5). It is worth mentioning that the libraries need to be defined based on a previous project/circuit block diagram study, with associated member responsibilities. In the case study, each one of the students was associate to a library, and also the coordinator, as he/she may also act as a designer. Tasks were defined, with priority and conclusion date. Each participant needs to annotate how the work is doing, so as to provide the coordinator or professor well informed.

¹The coordination function is temporal. In other words, any participant eventually can become a coordinator.

After the libraries were finished, (simulated schematics with associated symbols and pins, and layout), the students committed their libraries, twice, to check for the library upload and versioning (in their respective local repositories). This operation was performed via the Commit button, visible in Fig. 5) With the local repositories filled in and updated, the coordinator compiled (via Generate MAIN REPOSITORY button, visible in Fig. 4) the project, copying the libraries from the N users local repositories to the main one, having all the RINGOSCILLATOR project libraries available for design finishing. Visualization of compiled libraries are possible via Manage MAIN REPOSITORY button). Had the coordinator found any library missing or problem, he/she may had triggered an alert (through the communication system) to the right student, with proper instructions. Fig. 6 shows the editing of the ring oscillator (with attached electrical simulation), after compilation, in the main repository.

A. Results

Evaluation based on the case study has revealed the following (most of them were observed and agreed among the participants):

- Overall design organization is greatly improved, as the collaborative approach requires a coordinated work. The student is committed with his/her task, and will take care of providing documentation, and having his/her files updated and reliable;
- Design safety is greatly enhanced. The collaborative environment, as described and implemented here, provides main, local and version repositories. Whereas this is mainly thought to support collaboration, it also serves as a backup system for those files committed and compiled;
- Design speed-up, not during the individual designs, but

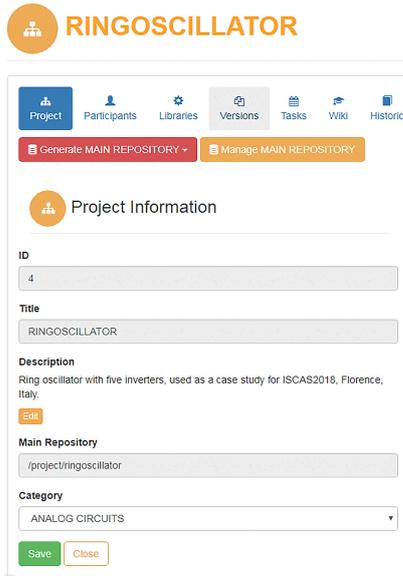


Fig. 4. IC-Collab Environment Project-screen.

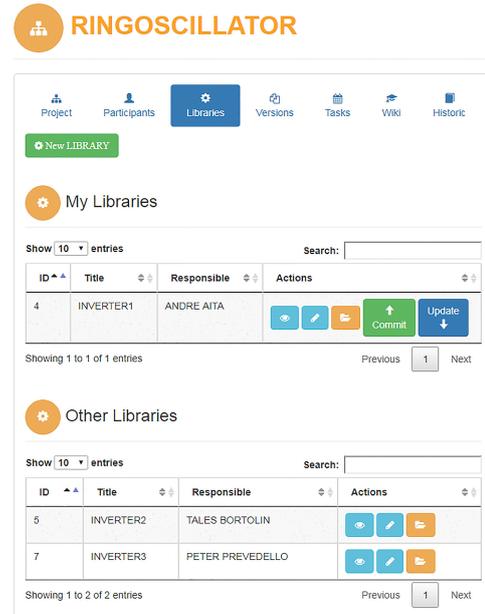


Fig. 5. IC-Collab Environment Libraries-screen.

when building the final project, due to improved organization;

- Increased quality of documentation. Under the collaboration environment, documentation needs to be delivered in a regular basis. If the project coordinator realizes the documentation is poor, or missing, he/she may trigger an alert to the student. Thus, at the time of the final compilation, most of the design documentation is available;
- Online collaboration and immediate communication among members is available within the environment as a premise for a virtual collaboration environment, which eases the whole design.

As the case study was preliminary, future evaluations with more complex designs in a first regular analog IC design course are underway.

V. CONCLUSION

This paper described a virtual collaborative environment to support teamwork-based education of analog IC design. The environment was planned to cope with those issues often observed during the design of such circuits in academic courses as lack of collaboration among students, lack of organization and documentation. With the system developed here, means for basic collaboration among students and coordinator/professor, versioning and documenting of repositories are made available with minimum efforts. Based on the results of a case study, design improvements in several aspects like higher overall design organization, documentation, safety were observed. As a final improvement, design libraries are easily made available for future courses by means of local shared student-libraries repositories.

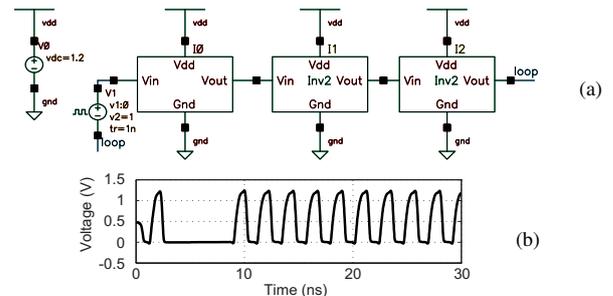


Fig. 6. (a) Schematic of the ring oscillator, after compilation, and (b) electrical simulation.

REFERENCES

- [1] Dehaene, W., *et al.*, "Circuits and Systems Engineering Education through Interdisciplinary Team-based Design Projects," *Proc. of IEEE Int. Symposium on Circuits and Systems (ISCAS)*, pp. 1195–1198, 2011.
- [2] De la Rosa, J. M., "A Teamwork-based Education Strategy for Teaching Lab of Analog Integrated Circuits Design," *Proc. of IEEE Global Engineering Education Conference (EDUCON)*, pp. 1–4, 2012.
- [3] Fernandes, J. R., "An Analog Integrated Systems Course Proposal for a 2nd Cycle of Studies of the Bologna Process," *Proc. IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 133–136, 2016.
- [4] Maróti, M., *et al.*, "Online Collaborative Environment for Designing Complex Computational Systems", *Proc. 14th International Conference on Computational Science (ICCS), Procedia Computer Science*, v. 29, pp.2432–2441, 2014.
- [5] Ellis, C. A., Gibbs, S. J., and Rein, G.L., "Groupware: some issues and experiences," *Communications of the ACM*, v.34, n.1, pp. 39–58, Jan. 1991.
- [6] Fuks, H., *et al.*, "Applying the 3C Model to Groupware Development," *International Journal of Cooperative Information Systems (IJCIS)*, v.14, n.2-3, pp. 299–328, Jun-Sep. 2005.
- [7] Fuks, H., *et al.*, "The 3C Collaboration Model," *Encyclopedia of E-Collaboration*, Kock, N., Ed., New York: Information Science Reference, pp. 637–644, 2007.
- [8] Welling, L., and Thomson, L., *PHP and MySQL Web Development*. 4th Ed., 2009. Pearson Education Inc.
- [9] GIT. Available on-line: <https://git-scm.com>, 2017.