Eduardo Marcate Garcia dos Anjos

Relatório de Estágio na Coordenação de Laboratórios Didáticos de Informática

Eduardo Marcate Garcia dos Anjos

Relatório de Estágio na Coordenação de Laboratórios Didáticos de Informática

Relatório de Estágio apresentado ao Curso de Graduação em Ciência da Computação como requisito parcial para obtenção do grau de Bacharel.

Universidade Federal do ABC - UFABC Graduação

Orientador: Prof^a Dr^a Debora Maria Rossi de Medeiros

Brasil

2018

Eduardo Marcate Garcia dos Anjos

Relatório de Estágio na Coordenação de Laboratórios Didáticos de Informática

Relatório de Estágio apresentado ao Curso de Graduação em Ciência da Computação como requisito parcial para obtenção do grau de Bacharel.

Prof^a Dr^a Debora Maria Rossi de Medeiros Orientador

Aderbal de Moraes Jr. SIAPE 1827087 - Supervisor

Brasil 2018



Resumo

O presente relatório refere-se ao trabalho realizado na Coordenação de Laboratórios Didáticos de Informática (CLDI) no âmbito das disciplinas Estágio Supervisionado I, II e III realizado para o curso de Ciência da Computação da Universidade Federal do ABC durante o período compreendido entre os dias 03 e 19 de maio de 2018 (I), entre os dias 13 de junho e 07 de julho de 2018 (II) e entre os dias 01 e 25 de agosto de 2018 (III). Neste documento são descritas as características da universidade, além das atividades realizadas e conhecimentos adquiridos durante o período de estágio. É apresentado, além do trabalho cotidiano, o desenvolvimento de dois sistemas web, que apesar de estarem fora das atribuições do cargo, fazem parte de um conjunto de demandas levantadas para auxiliar e melhorar o funcionamento da universidade. Também é descrita a aplicação dos conceitos obtidos em sala de aula tanto na execução das tarefas cotidianas do setor quanto no desenvolvimento da versão web do sistema de controle de patrimônio e do sistema de solicitação de software.

Palavras-chaves: Técnico em Informática. Gestão de Laboratórios. Suporte. Sistema de Patrimônio. Sistema de Solicitação de Software.

Lista de ilustrações

| Figura 1 — Sistema para solicitação de $software$ antes do início do quadrimestre | 5 |
|---|----|
| Figura 2 – Lista dos <i>software</i> solicitados através do sistema | 6 |
| Figura 3 $-$ Servidor $N\!F\!S$ contendo as imagens de cada laboratório da universidade | 7 |
| Figura 4 – Instalação de imagem através do sistema software Clonezilla | 8 |
| Figura 5 – Opções avançadas do software Clonezilla | 9 |
| Figura 6 – Inicialização do $software\ Clonezilla$ com parâmetros pré-selecionados . | 9 |
| Figura 7 – Agenda de atividades dos laboratórios do <i>campus</i> Santo André | 11 |
| Figura 8 – Versão online da agenda de atividades dos laboratórios do campus | |
| Santo André. | 11 |
| Figura 9 — Central de Serviços - reserva de laboratório de informática | 12 |
| Figura 10 – Diagrama Enhanced entity–relationship | 14 |
| Figura 11 — Adicionar um novo local a ser gerenciado pela equipe | 15 |
| Figura 12 – Excluir um local gerenciado pela equipe | 15 |
| Figura 13 — Adicionar um novo objeto a ser gerenciado pela equipe, vinculado a um | |
| local | 16 |
| Figura 14 – Excluir um objeto gerenciado pela equipe | 17 |
| Figura 15 – Movimentação de um objeto entre locais | 17 |
| Figura 16 – Relatório de uma conferência dos objeto de um local | 18 |
| Figura 17 – Lista de objetos vinculados a um local | 19 |
| Figura 18 – Histórico de um objeto | 19 |
| Figura 19 – Listagem de software solicitados | 21 |
| Figura 20 — Adicionando dados gerais de uma nova solicitação | 21 |
| Figura 21 — Adicionando detalhes de uma nova solicitação | 22 |
| Figura 22 - Procedimento de instalação ou teste de um software | 22 |

Sumário

| 1 | Intr | odução | 1 |
|----|--------|--|----|
| 2 | Loc | al de Trabalho | 3 |
| | 2.1 | Criação da UFABC | 3 |
| | 2.2 | Panorama Local | 3 |
| | 2.3 | Setor onde o estágio é realizado | 4 |
| 3 | Ativ | vidades desenvolvidas | 5 |
| | 3.1 | Preparação dos laboratórios para o início do quadrimestre | 5 |
| | | 3.1.1 Solicitação de software | Ę |
| | | 3.1.2 Preparação das imagens de disco | 6 |
| | | 3.1.3 Configurações pós imagem | 10 |
| | 3.2 | Controle de acesso aos laboratórios | 10 |
| | 3.3 | Desenvolvimento de um sistema $online$ para controle de patrimônio | 12 |
| | | 3.3.1 Ferramentas Utilizadas | 12 |
| | | 3.3.2 Desenvolvimento | 13 |
| | | 3.3.2.1 Banco de Dados | 13 |
| | | 3.3.2.2 Criação do Sistema Web | 14 |
| | | 3.3.3 Teste com usuários e modificações | 19 |
| | 3.4 | Desenvolvimento de um sistema online para solicitação de software | 20 |
| | 3.5 | Implantação de Software | 23 |
| | 3.6 | Diagnóstico de problema em <i>hardware</i> específico | 24 |
| 4 | Con | nhecimento aplicado | 25 |
| 5 | Sug | gestões de trabalhos futuros | 29 |
| 6 | Con | nclusão | 31 |
| | | | |
| Re | eferêi | ncias | 33 |
| A | nexo | os | 35 |
| ΔΙ | NFX | O A Plano de Atividades | 37 |

1 Introdução

Instituida em 2005 pela lei nº 11.145 de 26/07/2005, A Fundação Universidade Federal do ABC (UFABC) iniciou suas atividades em 2006 com um campus em Santo André e desde o começo tem o propósito de explorar abordagens inovadoras, tanto em pesquisa quanto no ensino. A UFABC foi a primeira Universidade a ofertar os cursos Bacharel em Ciência e Tecnologia (BCT) e Bacharel em Ciências e Humanidades (BCH)(UFABC, 2017a). Ao contrário de muitas outras instituições de ensino superior, a UFABC não é constituída por departamentos, mas sim por três grandes centros: o Centro de Ciências Naturais e Humanas, o Centro de Matemática, Computação e Cognição e o Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas, incentivando o desenvolvimento de atividades interdisciplinares. A universidade possui atualmente 700 docentes, todos com título de doutor, e 759 técnicos administrativos, cuja formação varia entre nível médio e doutorado, além de 45 estagiários e 172 funcionários terceirizados para atividades de suporte (UFABC, 2017d).

Atualmente a universidade conta com dois *campi*, localizados em Santo André e São Bernardo do Campo, totalizando mais de 110 mil metros quadrados de área construída. Desde 2010 o ingresso é realizado através do Exame Nacional do Ensino Médio (ENEM) e os candidatos escolhem no momento da inscrição um dos dois possíveis cursos de entrada: BCT ou BCH (UFABC, 2017b). Há 24 cursos de formação específica que os alunos podem optar após os bacharelados interdisciplinares, de acordo com o curso escolhido para o ingresso (UFABC, 2017a).

O curso de bacharelado em Ciência da Computação, por exemplo, tem como prérequisito de ingresso a conclusão do BCT. Dentre as disciplinas obrigatórias desse curso há a de Estágio Supervisionado, na qual o aluno, sob supervisão de um professor vinculado ao programa, deve realizar um período de trabalho relevante ao curso e produzir um relatório descrevendo as atividades. O presente relatório destina-se a cumprir parte dos requisitos desta disciplina e descreve as atividades realizadas como funcionário na própria instituição, detalhando as atividades realizadas dentro das atribuições como técnico de laboratório e o desenvolvimento de ferramentas para automatizar tarefas pertinentes à rotina de trabalho. O estágio teve duração de 100 horas e foi realizado no cargo de Técnico Laboratório de Informática no campus de Santo André.

O restante desse documento está organizado da seguinte maneira: o Capítulo 2 apresenta informações sobre a criação da universidade e sobre o local de trabalho; o Capítulo 3 discorre sobre as atividades desenvolvidas durante o período; o Capítulo 4 detalha como o conhecimento obtido durante o curso foi aplicado; o Capítulo 5 apresenta

de maneira sucinta as sugestões de trabalhos futuros e o Capítulo 6 apresenta a conclusão do relatório.

2 Local de Trabalho

Neste capítulo serão apresentadas breves informações quanto a criação da universidade, o panorama local no qual a universidade está inserida e o local de trabalho.

2.1 Criação da UFABC

Em 2004 o Ministério da Educação encaminhou ao Congresso Nacional o Projeto de Lei nº 3962/2004 que previa a criação da Universidade Federal do ABC refletindo uma vontade antiga do Governo Federal em implantar uma instituição de nível superior na região do Grande ABC. Após aprovação no legislativo, a lei foi sancionada e publicada no Diário Oficial da União de 27 de julho de 2005. Pensada desde o início para ser uma universidade inovadora e considerando as mudanças que ocorrem no campo da ciência, o Projeto Acadêmico da UFABC propõe uma matriz interdisciplinar, inédita no país, caracterizada pela intercessão de várias áreas do conhecimento científico e tecnológico (UFABC, 2017c). Por ter iniciado as atividades já com essa proposta, o comitê de implantação da universidade tinha grande liberdade para quebrar alguns paradigmas encontrado no ensino superior nacional, dessa forma o projeto pedagógico encoraja os alunos a se tornarem responsáveis por suas próprias vidas em vez de apenas escutarem o que devem fazer (UFABC, 2017b).

O projeto da UFABC, não só considera a criação de um ambiente acadêmico favorável ao desenvolvimento social mas também contribui para a busca de soluções para os problemas regionais e nacionais e para isso coopera com outras instituições de ensino e pesquisa, no Brasil e no exterior, além de instâncias do setor industrial e os três poderes da república (UFABC, 2017c).

2.2 Panorama Local

Até a chegada da UFABC em 2005/2006, as cidades de Santo André, São Bernardo do Campo, São Caetano, Diadema, Mauá, Ribeirão Pires e Rio Grande da Serra, apresentavam uma enorme demanda de vagas no ensino público superior, sendo que somente a região do ABC possuía mais de 2,5 milhões de habitantes. A oferta de vagas de ensino superior era de 45 mil vagas, sendo em sua maioria de instituições privadas. Dos cerca de 77 mil estudantes de nível superior, cerca de 65% estavam em instituições privadas. Outro ponto a se considerar é que uma pequena parte das instituições se dedicam a pesquisa além do ensino e poucas empresas no setor de tecnologia e engenharia apresentam investimentos em pesquisa aplicada (UFABC, 2017c).

2.3 Setor onde o estágio é realizado

A Coordenação de Laboratórios Didáticos de Informática (CLDI) é vinculada a Pró-Reitoria de Graduação (PROGRAD) e é constituída atualmente por um técnico de assuntos educacionais na função de coordenador, e 16 técnicos de informática e 1 tecnólogo de geoprocessamento. Essa coordenação é responsável pela manutenção e suporte de 23 laboratórios de informática, 2 laboratórios de praticas de ensino e 2 laboratórios de geoprocessamento, distribuídos nos dois *campi* da UFABC. Aproximadamente 900 computadores estão sob responsabilidade da CLDI, além de diversos equipamentos de rede e 2 servidores dedicados a sistemas internos.

3 Atividades desenvolvidas

Neste capítulo serão apresentadas informações relevantes sobre as atividades desenvolvidas durante o período de estágio.

3.1 Preparação dos laboratórios para o início do quadrimestre

Para que os laboratórios possam atender adequadamente as demandas das atividades acadêmicas, inicia-se aproximadamente 4 semanas antes do fim de cada quadrimestre o procedimento de preparação das imagens dos discos dos computadores, esse procedimento é dividido em etapas, apresentadas nas Seções 3.1.1, 3.1.2 e 3.1.3.

3.1.1 Solicitação de software.

Um sistema próprio, desenvolvido em conjunto com outros técnicos exclusivamente para essa finalidade e atualmente sob responsabilidade da PROGRAD permite que os professores responsáveis por cada disciplina solicitem os software necessários para ministrar as aulas durante o quadrimestre seguinte. Fazendo a seleção a partir de uma lista predefinida de software livres ou cuja universidade possui licença para uso, ou ainda, adicionar um não listado (desde que com licença livre). Para acessar o site, o professor precisa utilizar seu login institucional, dessa forma cada solicitação é vinculada ao autor. A Figura 1 apresenta o sistema utilizado no procedimento:



Figura 1 – Sistema para solicitação de software antes do início do quadrimestre

O sistema permite ainda geração de um relatório apresentando todas solicitações registradas, permitindo aos usuários a conferência das informações, a Figura 2 apresenta um exemplo de relatório.

| | Solicitação de softwares | | | | | |
|--|--|---|---|---|-----------------------------------|--|
| F | Relatório: Solicita | ação de Softwares - 2º | quadrimestre de 2017 | | | |
| Fo | Formulário de Solicitação Gerar Excel | | | | | |
| Softwares solicitados Observações adicionais | | | | | | |
| | Professor | Disciplina | Software | Versão | Observações | |
| | AMAURY KRUEL BUDRI | ESTI001-13 - Programação de Software Embarcado | Argo UML and de Janela http://argouml-downloads.tigris.org/nonav/argouml- 0.34/ArgoUML-0.34-setup.exe | Versão: 0.2 Plataforma: Windows | | |
| | AMAURY KRUEL BUDRI | ESTI001-13 - Programação de Software Embarcado | Eclipse | Versão: Plataforma: Windows | Eclipse CDT para c/C++ | |
| | AMAURY KRUEL BUDRI | ESTI001-13 - Programação de Software Embarcado | MinGW | Versão: 5.1.6 Plataforma: | | |
| | AMAURY KRUEL BUDRI | ESTI001-13 - Programação de Software Embarcado | Oracle Virtualbox https://www.virtualbox.org/wiki/Downloads | Versão: 4.3.14 Plataforma: Windows | | |
| | AMAURY KRUEL BUDRI | ESTI014-13 - Telefonia Fixa Moderna | Freeswitch | Versão: Plataforma: Linux | Apenas na máquina do professor | |
| | AMAURY KRUEL BUDRI | ESTI014-13 - Telefonia Fixa Moderna | Oracle Virtualbox https://www.virtualbox.org/wiki/Downloads | Versão: 4.3.14 Plataforma: Windows | | |
| | AMAURY KRUEL BUDRI | ESTI014-13 - Telefonia Fixa Moderna | PJSUA | Versão: 2.1 Plataforma: | | |

Figura 2 – Lista dos software solicitados através do sistema

De posse deste relatório e da planilha de alocações de disciplinas fornecida pela PROGRAD, foi feito o cruzamento das informações permitindo saber quais software foram solicitados em cada laboratório. Durante esta etapa foi feita uma verificação de compatibilidade entre os requerimentos computacionais de cada software e a capacidade dos computadores de cada laboratório, uma vez que diferentes laboratórios podem possuir diferentes arquiteturas computacionais. Algumas disciplinas foram realocadas para laboratórios que melhor atendem as demandas. Finalizada essa verificação realizou-se a compilação de uma lista de software necessários em cada laboratório.

3.1.2 Preparação das imagens de disco.

A lista compilada de solicitações de software (Seção 3.1.1) é, então, anexada a um chamado feito ao Núcleo de Tecnologia da Informação (NTI) setor da universidade que é responsável por gerenciar as licenças de software bem como os computadores nos quais eles serão instalados. O NTI utiliza um computador que tem exatamente a mesma configuração dos computadores dos laboratórios e faz a instalação e testes preliminares, e então gera uma imagem do disco rígido utilizando o software livre Clonezilla. Esta ferramenta é utilizada pois reduz o tamanho do arquivo final, permitindo a cópia apenas

do espaço utilizado no disco (CLONEZILLA, 2017). Essa imagem é então disponibilizada em um servidor *Network File System (NFS)* específico como mostrado na Figura 3.

```
😑 🗊 ufabc@ufabcdell9010:/home/partimag
RQUIMEDES-W-L506_1-2017-03-23-EYU-V17.1-img
                                                                       HP6300-D-TA_PROF-2017-03-24-EYU-V17.1-img
ELL7010-W10-AULA-2017-05-15-EYU-V17.2-img
                                                                       HP6300-W-SALA-2016-01-25-EYU-V2016Q1-img
DELL7010-W10-AULA_LOUSA-2016-12-23-EYU-V17.1-img
DELL7010-W10-AULA_LOUSA-2017-05-16-EYU-V17.2-img
                                                                       HP6300-W-TA_PROF-2016-06-20-EYU-V16.2-img
DELL7010-W10-TA_PROF-2016-12-14-EYU-V16.3-img
DELL7010-W8.1-CLEAN-2016-12-12-EYU-V16.3-img
DELL7010-W8.1-TA_PROF-2016-12-12-EYU-V16.3-img
                                                                       HP6305-D-AULA-2016-11-17-EYU-V16.3-img
                                                                       HP6305-W-CLEAN-2016-11-09-EYU-V16.3-img
                                                                       HP6305-W-EMPRESTIMO-2016-11-07-EYU-V16.3-img
)ELL7010-W8TEST-2016-02-24-FDCI
)ELL740-W-CLEAN-2017-05-02-EYU-V17.1-img
                                                                       HP6305-W-TA_PR0F-2016-11-09-EYU-V16.3-img
HP6360B-W-ACESSIBILIDADE-2016-08-17-EYU-V16.2-img
                                                                       HP6360B-W-TA_PROF-2016-08-16-EYU-V16.2-img
ELL780-W-AULA-2017-02-09-EYU-V17.1-img
DELL780-W-AULA_LOUSA-2016-09-08-EYU-V16.3-img
DELL780-W-BIBLIO-2016-03-01-EYU-V16.1-img
                                                                       HP6470b-W-TA-2017-02-22-FDCI
DELL780-W-B0-2016-09-27-EYU-V16.3-img
DELL780-W-CLEAN-2017-04-20-EYU-V17.1-img
                                                                       HP8100-D-LX03B-2017-05-22-EYU-V17.2-img
ELL780-W-DOACA0-2017-04-20-EYU-V17.1-img
                                                                       HP8100-W-CLEAN-2016-09-15-EYU-V16.3-img
                                                                       HP8100-W-L410_1-2016-09-30-EYU-V16.3-img
ELL780-W-ESTUDO-2016-12-22-V17.1-img
ELL780-W-TA_PROF-2016-06-20-EYU-V16.2-img
                                                                       HP8100-W-TA_PROF-2016-12-06-EYU-V16.3-img
ELL9010-D-L403_2_TREINA-2017-05-03-FDCI-V17.1C
DELL9010-D-L40X_2-2017-05-23-EYU-V17.2-img
DELL9010-D-L50XB-2017-05-25-EYU-V17.2-img
                                                                       HP8200-W-PROPG-2016-05-06-EYU-V16.2-img
HP8200-W-TA_PROF-2016-12-08-EYU-V16.3-img
HP_PROBOOK4440S-W-TA_PROF-2016-12-23-EYU-V16.3-img
DELL9010-D-LABSECO-2016-06-02-EYU-16.2-img
DELL9010-D-SBC-A2-L001-2016-09-09-EYU-V16.3-img
ELL9010-D-SBC-LAB-2016-09-12-EYU-V16.3-img
ELL9010-L404A2-ubuntu-opencv-17.06.16-img
                                                                       HP_Z420-D-L401_1-2017-05-10-EYU-V17.2-img
HP_Z420-D-L401_1-2017-05-24-EYU-V17.2-img
ELL9010-W-CLEAN-2017-02-23-EYU-V17.1-img
ELL9010-W-L50X_1-2016-01-22-EYU-V2016Q1-img
DELL9010-W-LABSECO-2016-06-02-EYU-V16.2-img
                                                                       HP_Z420-W-GE0-2017-05-18-EYU-V17.2-img
HP_Z420-W-GE0-2017-05-26-EYU-V17.2-img
ELL9010-W-PROPG-2016-05-06-EYU-V16.2-img
                                                                       Imagens_Labs.txt
DELL9010-W-TA_PROF-2016-06-23-EYU-V16.2B-img
DELL_T3500_2013-ing
DELL_T3500_2017-ing
DELL_VOSTRO_3500-W-TA_PROF-2016-08-30-EYU-V16.2-ing
                                                                       POSITIVO MOBO-W-EMPRESTIMO-2017-04-27-EYU-V17.1-img
```

Figura 3 – Servidor NFS contendo as imagens de cada laboratório da universidade

Uma vez que a imagem é disponibilizada no servidor, utiliza-se novamente o *Clonezilla* para copiar essa imagem para o disco de um computador do laboratório para realização de novos testes desta vez pelos técnicos da CLDI, qualquer erro encontrado é reportado ao NTI para correção e o procedimento recomeça. Isso foi realizado para cada laboratório. Apenas dois laboratórios são exceções, o L402-2 e o L408-2, por possuírem arquitetura especifica e não terem sido adquiridos através de licitações geridas pelo NTI tem imagem gerada pela própria CLDI. Como esses laboratórios compartilham a mesma arquitetura aplica-se a mesma imagem em ambos.

Atendendo uma demanda apresentada por professores para o terceiro quadrimestre de 2018 a imagem dos laboratórios L402-2 e L408-2 foi totalmente refeita, já utilizando-se versões atualizadas dos *software* ali presentes. Essa atividade demandou grande esforço uma vez que estes laboratórios são utilizados para algumas disciplinas com demandas específicas tais como Programação Paralela, Computação Gráfica, Modelagem Molecular e Aplicações de Microcontroladores, entre outras, as quais necessitam de configurações diferenciadas. Ao todo o procedimento de instalação, configuração e testes levou cerca de três semanas devido à demandas de diferentes disciplinas que precisaram ser conciliadas.

Após a finalização de todos os testes, a imagem é copiada simultaneamente para todos computadores do laboratório, garantindo uniformidade na configuração de cada máquina. Esse procedimento foi realizado respeitando uma escala previamente definida. Esta medida foi tomada para evitar sobrecarga tanto no servidor de imagem quanto na rede da UFABC, o que poderia resultar em falhas durante o processo de cópia e gerando um aumento no tempo para conclusão desta atividade. Uma tela desse procedimento pode ser vista na Figura 4.

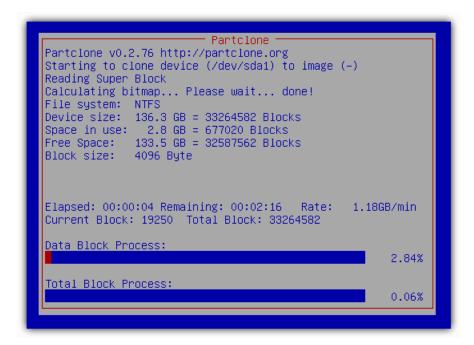


Figura 4 – Instalação de imagem através do sistema software Clonezilla

Devido às características dos laboratórios é necessário utilizar alguns parâmetros adicionais durante a execução do *Clonezilla*, conforme mostrado na Figura 5. As opções criticas são "-g auto" e "-hn1 PC", responsáveis por forçar a reinstalação do GRUB¹ e renomear o computador com um nome único, respectivamente. Isso é necessário para que o computador possa iniciar os sistemas operacionais instalados (*MS Windows* e *Ubuntu Linux*) e para que o computador possa ser registrado no servidor de domínio.

¹ GRand Unifield Bootloader - um gerenciador de inicialização livre

```
NCHC Free Software Labs, Taiwana advanced extra parameters | Mode: restoredisk |

Set advanced parameters (multiple choices available). If you have no idea, keep the default values and do NOT change anything. Just press Enter. (Press space key to mark your selection. An asterisk (*) will be shown when the selection is done)

[**] -g auto Reinstall grub in client disk MBR (only if grub config exists)
[**] -e1 auto Automatically adjust filesystem geometry for a NTFS boot partition if exists [**] -e2 sfdisk uses CHS of hard drive from EDO(for non-grub boot loader)
[**] -nogui Use text output only, no TUI/GUI output
[**] -hno PC Change MS Win hostname (based on IP address) after clone
[**] -hni PC Change MS Win hostname (based on IP address) after clone
[**] -v Prints verbose messages (especially for udpcast)
[**] -e Client waits for confirmation before cloning
[**] -t Client does not restore the MBR (Master Boot Record)
[**] -t1 Client does not restore the MBR (Master Boot Record)
[**] -t1 Client restores the prebulit bootloader from syslinux (For Windows only)
[**] -r Try to resize the filesystem to fit partition size
[**] -e sfdisk uses the CHS value of hard drive from the saved image
[**] -icrc Ignore CRC checking of partclone
[**] -irr Do not remove Linux udev hardware record after restoring.
[**] -iisis Do not update syslinux-related files after restoring.
[**] -iisis Do not update syslinux-related files after restoring
[**] -iicfi Skip checking destination disk size before creating partition table
[**] -iifi MRITTE MBR (512 B) again after Image is restored. Not OK for partition table diffe
[**] -j2 Clone the hidden data between MBR and 1st partition
[**] -m Check image by MDS checksums
[**] -en Check image by MDS checksums
[**] -a Do NOT force to turn on HD OMA
[**] -on Run script in $00S_POSTRUN_DIR before clone starts
[**] -srel Save restoring error log in image dir
```

Figura 5 – Opções avançadas do software Clonezilla

Para acelerar o processo e evitar erros, configurou-se um *pen-drive* para que iniciasse o *Clonezilla* automaticamente com os parâmetros opcionais já selecionados para cada laboratório, como pode ser visto na Figura 6.

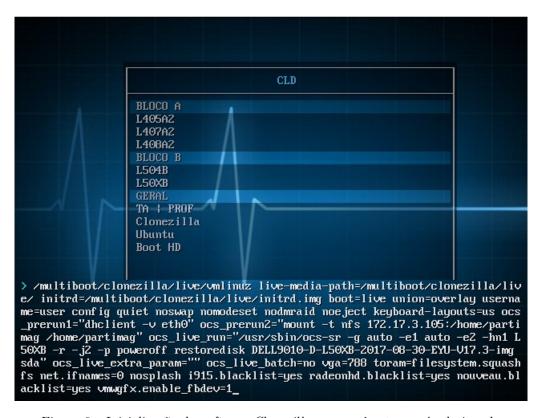


Figura 6 – Inicialização do $software\ Clonezilla$ com parâmetros pré-selecionados

3.1.3 Configurações pós imagem

Cada computador precisa ser registrado no servidor de domínio após a restauração do sistema, permitindo aos usuários acesso através de usuário e senha próprios. Devido ao grande número de computadores foi criado pelo autor do presente documento juntamente com dois outros técnicos um script que pode ser executado dentro da rede local e que faz esse processo automaticamente, bastando fornecer uma lista de endereços IP² dos computadores de cada laboratório. Essa lista precisa ser gerada no mesmo dia da execução do script pois devido à politicas de rede do NTI, o endereço dos computadores é renovado a cada 24 horas, não existindo qualquer garantia de que o endereço será mantido. Falhas de comunicação entre o computador e o roteador também podem levar a renovação do endereço. Esse script é executado em cada laboratório pela equipe técnica da CLDI, seguindo uma escala a fim de evitar sobrecarga do servidor de domínio, que em caso de falha impediria qualquer usuário da UFABC acessar qualquer computador com o sistema MS Windows. Por fim é realizada a configuração do projetor no computador designado ao professor em cada laboratório, em ambos sistemas operacionais.

3.2 Controle de acesso aos laboratórios

Além das alocações para aulas dos programas de graduação e de pós-graduação, atividades pontuais como treinamentos internos e atividades de extensão demandam acesso aos laboratórios. O controle dessas alocações é realizado através do *Google Agenda*, conforme pode ser visto na Figura 7.

² Internet Protocol

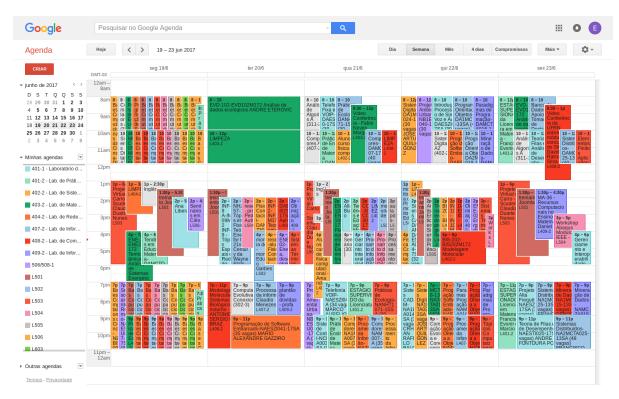


Figura 7 – Agenda de atividades dos laboratórios do campus Santo André.

Os usuários do laboratório tem acesso as mesmas informações através do página especifica no site da universidade³, mostrado na Figura 8.

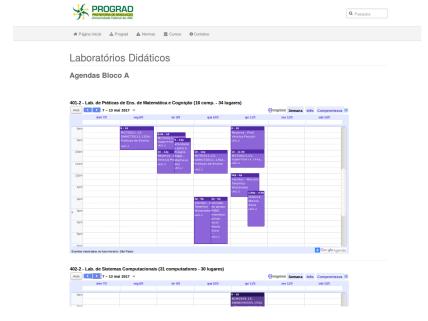


Figura 8 – Versão online da agenda de atividades dos laboratórios do campus Santo André.

Essa informação é importante pois os usuários podem solicitar reservas dos labora-

³ http://prograd.ufabc.edu.br/labs/1986

tórios para atividades através da Central de Serviços⁴, atualmente em fase de implantação. A interface é apresentada na Figura 9.

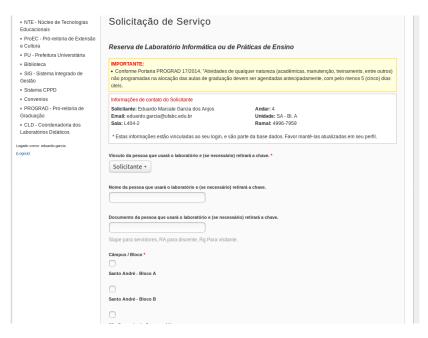


Figura 9 – Central de Serviços - reserva de laboratório de informática.

3.3 Desenvolvimento de um sistema *online* para controle de patrimônio

Durante o segundo quadrimestre de 2016, o setor foi submetido a um processo de auditoria interna, um dos problemas encontrados foi referente ao controle de inventário dos laboratórios. O controle de patrimônio utilizava apenas planilhas eletrônicas descentralizadas, tornando o processo de atualização lento e sujeito a erros, também dificultando a detecção de informações conflitantes, como um mesmo item sendo listado em dois laboratórios diferentes. Um software foi desenvolvido pela equipe técnica com o objetivo de sanar essas deficiências. Esse sistema não foi utilizado em ambiente de produção devido à características indesejadas, como falta de identificação e controle de usuário e dificuldade de acesso remoto, mas serviu de base para outro sistema desenvolvido pelo estagiário durante o período ao qual este relatório de estágio se refere.

3.3.1 Ferramentas Utilizadas

Após o levantamento e análise dos requisitos do projeto, foram selecionadas as seguintes ferramentas para o desenvolvimento do *software*.

⁴ http://central.ufabc.edu.br/

- MySQL Workbench: reúne diversas funcionalidades como construção visual, que permite criar modelos, gerar e gerenciar banco de dados e gerador de modelos Entidade-Relacionamento. Os recursos para desenvolvimento incluem criação, execução e otimização de consultas em Structured Query Language (SQL). Quanto a parte de administração é possível configurar servidores e contas de administradores, backup, restauração e auditoria dos dados (MYSQL, 2017).
- *PHP*: linguagem de *script* para servidores, trata-se de uma linguagem interpretada e de auto-nível, bastante difundida no mercado. É suportada pelos principais sistemas de servidor *web*, como o Apache⁵ utilizado na UFABC. Outra vantagem é o fato de facilmente se encontrar informações como tutoriais e livros de apoio, além de *Application Programming Interfaces (APIs)* e *frameworks*.
- NetBeans: este Integrated Development Environment (IDE) possui amplo suporte ao desenvolvimento de aplicações em PHP, o editor de códigos auxilia durante a codificação com dicas, formatação e refatoração. Há suporte a gerenciamento de projetos e recursos. O debugger permite utilizar algumas funcionalidades como pontos de interrupção e watches. Há controle de versões e integração com Git (NETBEANS, 2017).
- Zend Framework⁶: Framework PHP que pode ser integrada ao IDE Netbeans. Suporta testes de unidades, integração contínua e diferentes arquiteturas de software.

3.3.2 Desenvolvimento

Nas próximas seções estão listadas as etapas de desenvolvimento desta atividade e uma breve descrição do que foi realizado em cada uma delas.

3.3.2.1 Banco de Dados

Optou-se por manter a estrutura do banco de dados já existente, uma vez que não foram encontrados problemas ou limitações na modelagem original. O diagrama apresentado na Figura 10 mostra o modelo final. Observa-se que todas as chaves são numéricas, o que torna o gerenciamento e armazenamento de dados mais eficientes, uma vez que a descrição dos itens e dos estados dos mesmos aparecem diversas vezes, fazendo com que o banco de dados ocupe menor espaço se comparado com o tipo texto, além de ser uma boa prática na modelagem de bancos de dados.

⁵ https://www.apache.org/

⁶ https://framework.zend.com/

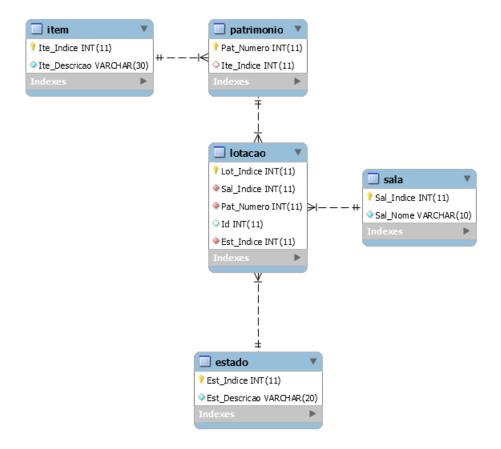


Figura 10 – Diagrama Enhanced entity-relationship

3.3.2.2 Criação do Sistema Web

As funções do sistema foram desenvolvidas gradativamente conforme o grau de complexidade das mesmas, cada função foi testada individualmente utilizando o framework de testes PHPunit⁷ através do framework Zend. Cada função foi integrada ao sistema e novos testes foram realizados com objetivo de certificar-se de que a integração ocorreu sem erros. A integração com o banco de dados foi realizada com o plugin Zend DB Adapter. Foram criadas classes para separar a parte lógica da partes que tratam com SQL através de mapeamento objeto-relacional. Para mitigar possiveis falhas de segurança do tipo SQL Injection, optou-se por utilizar Prepared Statements.

Uma das vantagens do framework Zend é a facilidade de trabalhar com a arquitetura Model-View-Controller (MVC), assim a interface com o usuário (front-end) pode ser desenvolvida em conjunto com as funções de back-end.

O sistema foi desenvolvido de maneira modular, utilizando controle de acesso integrado a outros sistemas da UFABC (OpenSSO) e com possibilidade de gerenciar permissões por módulo, assim o administrador do sistema pode autorizar determinados usuários a acessar apenas um subconjunto das informações e funções disponíveis.

https://phpunit.de/

Ao acessar o sistema o usuário autorizado pode cadastrar ou excluir um novo local a ser gerenciado (Figuras 11 e 12 respectivamente).

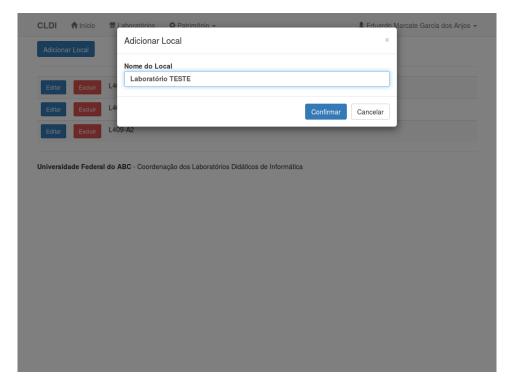


Figura 11 – Adicionar um novo local a ser gerenciado pela equipe

A função de exclusão é condicionada a não existência de conferências abertas para aquele local. Essa verificação ocorre para impedir inconsistências nos dados.

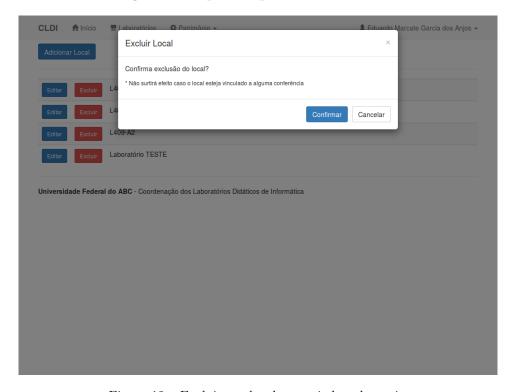


Figura 12 – Excluir um local gerenciado pela equipe

Também é possível cadastrar objetos patrimoniados, indicando o local onde o objeto ficará, assim como é possível excluir objetos cadastrados, com a mesma limitação da exclusão do local. Essas funções são mostradas nas Figuras 13 e 14 respectivamente.

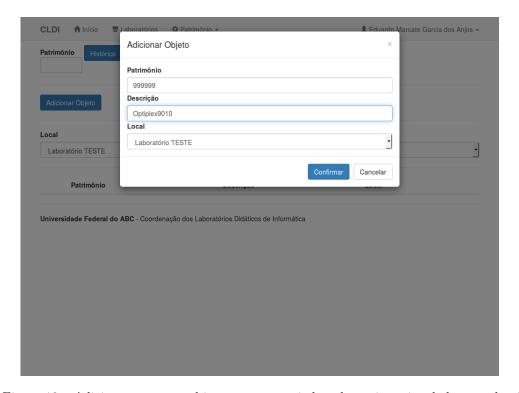


Figura 13 – Adicionar um novo objeto a ser gerenciado pela equipe, vinculado a um local

A função de criação de um objeto impede duplicidade de forma transparente ao usuário, verificando a existência do mesmo no sistema. O campo descrição tem recurso de sugestão de preenchimento.

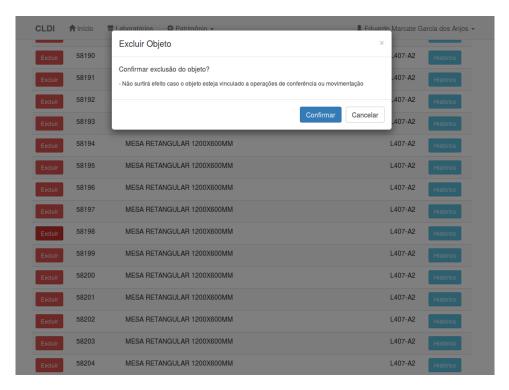


Figura 14 – Excluir um objeto gerenciado pela equipe

Também é possível realizar a movimentação do objeto entre locais, conforme apresentado na Figura 15.

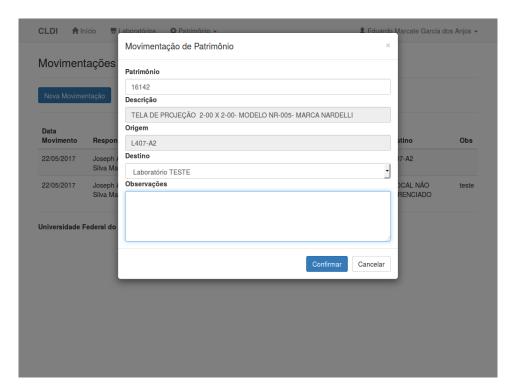


Figura 15 – Movimentação de um objeto entre locais

Rotineiramente é realizado um procedimento de conferência dos itens cadastrados,

o usuário pode indicar se o item foi encontrado em bom estado (OK), se não foi encontrado (NÃO ENCONTRADO) ou ainda fazer observações sobre o estado do item, a Figura 16 apresenta essa função.

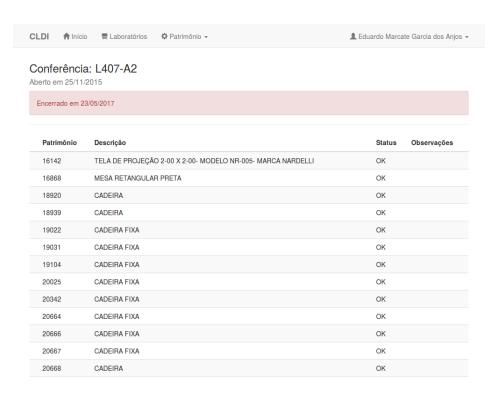


Figura 16 - Relatório de uma conferência dos objeto de um local

Por fim, é possível listar os objetos de um local, e visualizar o histórico de ações realizadas sobre um determinado objeto, conforme ilustram as Figuras 17 e 18.

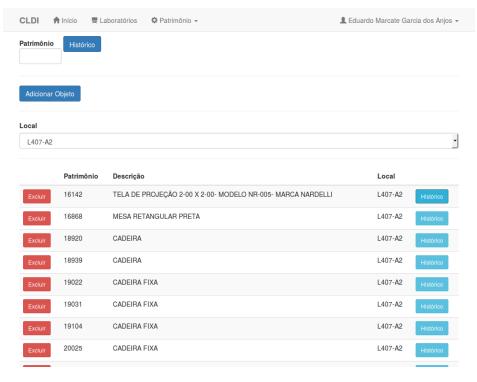


Figura 17 – Lista de objetos vinculados a um local

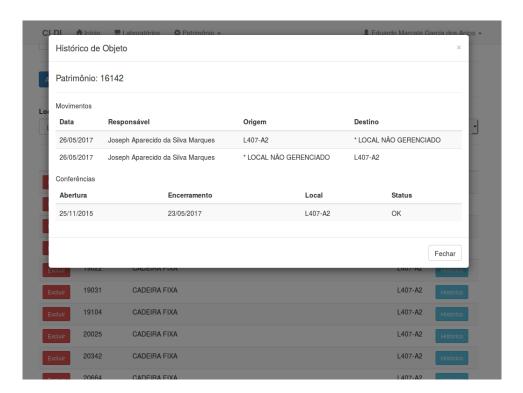


Figura 18 – Histórico de um objeto

3.3.3 Teste com usuários e modificações

Durante os testes com os usuários percebemos que algumas características indesejadas em funções do sistema desenvolvido, durante o cadastro de novos objetos, por exemplo, o usuário fornece a descrição do item e essa descrição pode tanto omitir diferenças fundamentais entre os itens quanto descrever de forma diferente itens iguais. O campo de texto livre foi substituído por um menu de seleção, evitando esse problema.

3.4 Desenvolvimento de um sistema *online* para solicitação de *soft-ware*

O procedimento descrito na Seção 3.1.1 é uma das principais fontes de reclamações por parte dos usuários dos laboratórios. Apesar de ser um processo bastante simples, alguns usuários fazem suposições falsas sobre o funcionamento interno do sistema, considerando que as solicitações para um determinado quadrimestre serão migradas para os quadrimestres seguintes. software de uso muito frequente acabam sendo instalados independentemente de haver solicitações devido ao know-how da equipe da CLDI, exigindo que as solicitações sejam revisadas por várias pessoas antes da compilação final da lista de software para cada laboratório. Visando mitigar esse problema uma nova versão para essa ferramenta está sendo desenvolvida com um fluxo de trabalho mais claro e com "memória" das solicitações anteriores. O sistema deverá compilar a lista automaticamente permitindo redução tanto da demanda de trabalho quanto dos prazos necessários para essa etapa.

Após o levantamento inicial dos requisitos optou-se por fazer uma análise do comportamento dos usuários no processo atual, coletando dados que serão utilizados tanto para guiar o desenvolvimento quanto como benchmark. As informações coletadas durante o processo de solicitação de software para o terceiro quadrimestre letivo de 2018 serão analisada e servirão para embasar o desenvolvimento do sistema. O framework Zend está sendo utilizando nesse sistema, permitindo a integração com os demais sistemas desenvolvidos pela CLDI. Atualmente o sistema conta com algumas funcionalidades implementadas em estado beta e o banco de dados de teste ainda não foi populado. Pretende-se popular o banco de dados de teste com informações oriundas do processo atualmente em uso, etapa na qual será possível avaliar a viabilidade da estrutura proposta e fazer as modificações necessárias para acomodar eventuais demandas não atendidas. A simplicidade para geração de interface de usuário no framework utilizado viabiliza a geração de telas mesmo nessa versão preliminar do sistema, permitindo a geração de mockups fiéis à realidade. Na Figura 19 é apresentada a interface da listagem de software.



Figura 19 – Listagem de software solicitados

As Figuras 20 e 21 apresentam a janela de adição de solicitação de software. Os campos foram pensados para guiar o solicitante a fornecer as informações mínimas necessárias para garantir o adequado atendimento das demandas de aula.

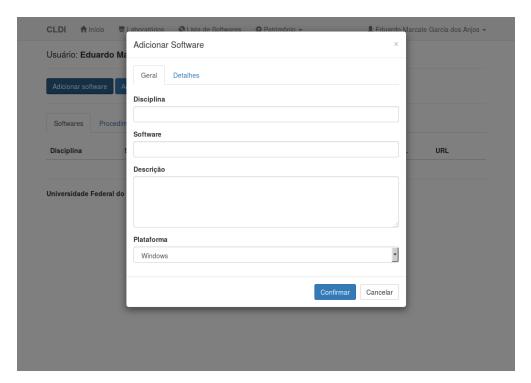


Figura 20 – Adicionando dados gerais de uma nova solicitação

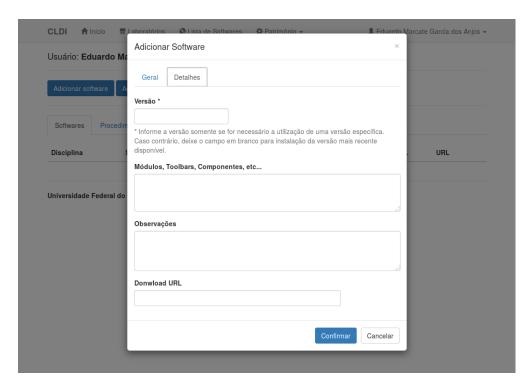


Figura 21 – Adicionando detalhes de uma nova solicitação

Adicionalmente é possível definir procedimentos padronizados de instalação e teste dos *software* solicitados, conforme apresentado na Figura 22.

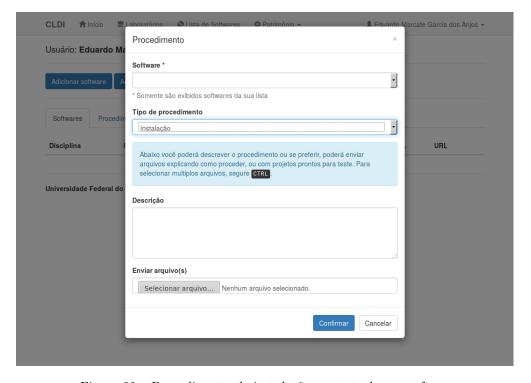


Figura 22 – Procedimento de instalação ou teste de um software

3.5 Implantação de Software

O desenvolvimento dos software foi realizado em servidor local acessível apenas pelos técnicos envolvidos no desenvolvimento. Após uma reunião com a chefia decidiu-se que o sistema de patrimônio estava maduro suficiente para ser migrado para o servidor de produção, onde poderá ser alimentado por todos usuários. Como os usuários estão espalhados nos dois campi verificou-se a necessidade de que o sistema colocado em um servidor acessível através de toda rede interna da instituição. Para atender à demanda de implantação, entrou-se em contato com o NTI para solicitar um servidor (máquina virtual) a ser administrado pela CLDI.

As configurações de *hardware* solicitadas foram:

• RAM: 4 GB

• Armazenamento: 50 GB

• SO: Debian 8

Quanto a questão de software que seriam necessários:

• Sistema de arquivos

• Servidor WEB Apache

• Servidor de banco de dados MySQL

Esse servidor será utilizado para outros sistemas que estão sendo desenvolvidos ou migrados pela equipe.

Após a análise do pedido pelos responsáveis do NTI, fomos atendidos com a seguinte configuração:

• RAM: 2 GB

• Armazenamento: 12 GB

• SO: CentOS

Foram recebidas recomendações de segurança do NTI, juntamente com usuário, senha, endereço IP e endereço de acesso para instalar os *software* base e prosseguir com a implantação do sistema. O sistema foi migrado para o servidor definitivo, onde foi avaliado pelos usuários e após sofrer pequenas modificações e correções de erros foi considerado pronto para uso.

O sistema de solicitação de softwares ainda não foi colocado no ambiente de produção.

3.6 Diagnóstico de problema em hardware específico

Com o objetivo de atender as demandas das disciplinas "programação paralela" da graduação em ciência da computação e "programação paralela e concorrente" da pósgraduação em ciência da computação existe no laboratório L408-2 uma máquina HP Z800, trata-se de um computador com dois processadores e duas placas gráficas de uso profissional fundamentais para atividades práticas das disciplinas. Durante o quadrimestre o computador começou a apresentar comportamento errático, deixando de funcionar completamente na terceira semana de junho. Um diagnóstico inicial indicava problemas nas placas de vídeo. Para determinar efetivamente qual era o defeito um experimento foi planejado utilizando a plataforma CUDA (Compute Unified Device Architecture). Um programa foi desenvolvido para gravar dados repetidamente na memória da placa de vídeo, permitindo identificar blocos defeituosos e assim foi possível isolar a placa defeituosa. Por trabalharem em conjunto através da tecnologia SLI (Scalable Link Interface) optou-se por substituir o conjunto completo.

4 Conhecimento aplicado

Abaixo estão discriminados os principais conhecimentos aplicados, referentes a cada disciplina, durante o período de estágio:

• Arquitetura de Computadores (STALLINGS; FIGUEIREDO; FIGUEIREDO, 2009):

O aprendizado obtido em sala de aula foi útil para melhor entendimento das diferenças entre as diferentes famílias e gerações de processadores, permitindo que a alocação de disciplinas exigentes em recursos computacionais fossem alocadas em um laboratório que, apesar de contar com uma arquitetura mais antiga, possui processadores mais velozes para operações mutithread. Também pude diagnosticar problemas de hardware e software de forma mais eficaz, identificando sintomas corretamente de acordo com conhecimento obtido na disciplina.

• Redes de Computadores (KUROSE; ROSS, 2010) e (TANENBAUM, 2003):

O entendimento geral do funcionamento da infraestrutura de rede me auxiliou a entender, na prática, como os problemas de congestionamento de rede acontecem e principalmente o que podemos fazer para evitá-los, sendo possível analisar o impacto na taxa de transferência na infraestrutura interna da UFABC ao conectar gradualmente computadores dos laboratórios ao servidor de imagem, o que acrescentou conhecimento prático a pontos que foram apresentados pontualmente nas aulas teóricas.

Conceitos como protocolos de conexão e seus tamanhos de quadro (ou *frames*) para transmissão de dados (carga útil), *multicast* e *broadcast* foram importantes para o entendimento e melhor aproveitamento dos recursos do *software Clonezilla*.

Utilizando o *software Advanced IP Scanner* verificou-se a existência de sub-redes em cada um dos laboratórios, o que auxiliou a realizar testes sistematizados em computadores para certificar-se de que os mesmo foram efetivamente inseridos no domínio.

Também pude notar que não existe qualquer estrutura de cache ou sincronização de pacotes, o que poderia tornar a conexão com o servidor de imagem muito mais eficiente, evitando assim o congestionamento da rede e oferecendo a possibilidade de que uma quantidade maior de computadores possa ser restaurada ao mesmo tempo.

• Sistemas Distribuídos (TANENBAUM; STEEN, 2007):

Além do desenvolvimento de um sistema cliente-servidor onde foram aplicados diversos conceitos visto em aula, também existe a interação com um servidor NFS

(que por características observadas trata-se de um *cluster*) através de máquinas clientes, utilizando o *software Clonezilla* para fazer a troca de dados.

Também foi possível verificar possíveis falhas de segurança em servidores utilizados para o desenvolvimento do trabalho.

• Sistemas operacionais (TANENBAUM, 2009):

O Clonezilla é um sistema construído para realizar tarefas específicas, sendo possível configurá-lo para atender às necessidades específicas da CLDI através da adição de parâmetros de execução. A utilização efetiva desse sistema só é possível com o entendimento de características dos sistemas operacionais que serão copiados. Conhecimentos da organização interna dos sistemas operacionais foram fundamenteis para preparação das imagens dos laboratórios L402-2 e L408-2, uma o sistema linux precisou ter seu kernel recompilado para melhor desempenho do driver de vídeo, fundamental para a disciplina de programação paralela e concorrente.

• Banco de dados (DATE, 2004):

Durante a análise do banco de dados que foi utilizado no sistema desenvolvido foram necessários conceitos de normalização de dados, construção do diagrama EER, escolha do tipo adequado de chave primária e chaves estrangeiras. Também foi utilizado a linguagem SQL, vista em aula, para criação do banco de dados, tabelas e especificações dos campos, realização de consultas reunindo informações de diversas tabelas (Join). A integração e conexão do banco de dados MySQL com o uma IDE para a construção da parte lógica do sistema foi similar ao visto em aula, com a diferença da linguagem usada no desenvolvimento ter sido PHP e não Java.

• Programação orientada a objetos (DEITEL; DEITEL, 2005):

Todos conceitos de programação orientada a objetos foram fundamentais para o desenvolvimento do *software*, utilizamos classes, objetos, herança, sobrecarga e aplicamos conceitos de projeto que foram vistos em sala, como encapsulamento e acoplamento fraco.

• Segurança de dados: (GOODRICH; TAMASSIA, 2013)

Prevenção de entrada de dados maliciosos por qualquer parte do *software*, como um ataque do tipo *SQL Injection*, por exemplo.

• Interface humano-máquina: (BENYON, 2011)

Alguns dos principais problemas apontados pelos usuários do sistema que está sendo substituído eram referentes a más escolhas de interface. Diversos conceitos da disciplina puderam ser aplicados, desde a escolha das cores até a realização de testes de usabilidade com usuários finais do sistema.

- Programação Paralela: (RAUBER; RÜNGER, 2013)
 Os conceitos e técnicas de programação paralela foram fundamentais para a criação de um experimento que possibilitou o diagnóstico de defeito em um sistema crítico.
- Análise de Algoritmos (DASGUPTA; PAPADIMITRIOU; VAZIRANI, 2006)
 O entendimento sobre eficácia e eficiência computacional, em termos tanto de processamento quanto memória, permitiu fazer escolhas mais acertadas a respeito da forma de implementação de funções específicas. Apesar de não ter realizado nenhuma prova de corretude os conceitos envolvidos foram úteis durante o processo de debugging dos sistemas implementados.

5 Sugestões de trabalhos futuros

A configuração do *Clonezilla* para iniciar automaticamente com parâmetros préselecionados é feita através da edição de um arquivo de texto, sendo comum a ocorrência de erros por parte do técnico. Esses erros podem gerar retrabalho e prejuízo às atividades. Pretendo desenvolver uma interface que simplifique essa edição, evitando equívocos.

Também estamos estudando a possibilidade de configurar um servidor DRBL¹ para permitir que o procedimento de imagem descrito na Seção 3.1.2 seja feito via *multi-cast*, enviando cada pacote de dados simultaneamente para todos computadores de uma rede virtual. Para funcionar, no entanto, esse sistema depende da liberação de trafego de pacotes especiais nos roteadores cuja administração é feita pelo NTI.

¹ Diskless Remote Boot in Linux (http://drbl.org/)

6 Conclusão

Trabalhar na área do curso me permite aplicar na prática conhecimentos adquiridos de maneira teórica, fixando conceitos e dando uma visão concreta de situações que durante o curso são apresentadas de forma abstrata, além de ter criado oportunidades para conhecer e utilizar novas ferramentas. Também notei a necessidade latente por ferramentas que ajudem a diminuir erros e evitem o retrabalho, garantindo uniformidade e previsibilidade na execução das tarefas.

O desenvolvimento do sistema online de controle de patrimônio, e mais recentemente do sistema de solicitação de *software* me proporcionou uma chance de aprender além da sala de aula, realizando desde o levantamento de requisitos até o desenvolvimento funcionalidades como controle de acesso, controle de permissões e interface gráfica, levando-me a ter maior confiança e independência no desenvolvimento de aplicações mais próximas a realidade.

Referências

BENYON, D. Interação humano-computador. Sao Paulo: Perason Prentice Hall, 2011. Citado na página 26.

CLONEZILLA. Sobre o Clonezilla. 2017. Disponível em: http://clonezilla.org/>. Citado na página 7.

DASGUPTA, S.; PAPADIMITRIOU, C. H.; VAZIRANI, U. *Algorithms*. [S.l.]: McGraw-Hill, Inc., 2006. Citado na página 27.

DATE, C. J. Introdução a sistemas de bancos de dados. [S.1.]: Elsevier Brasil, 2004. Citado na página 26.

DEITEL, H. M.; DEITEL, P. J. Java: Como Programar, 6a. Edição. [S.l.]: Pearson, 2005. Citado na página 26.

GOODRICH, M. T.; TAMASSIA, R. *Introdução à segurança de computadores*. [S.l.]: Bookman, 2013. Citado na página 26.

KUROSE, J. F.; ROSS, K. W. Redes de Computadores e a Internet. [S.l.: s.n.], 2010. Citado na página 25.

MYSQL. MySQL Workbench. 2017. Disponível em: http://www.mysql.com/products-/workbench/. Citado na página 13.

NETBEANS. NetBeans. 2017. Disponível em: https://netbeans.org/features-/index_pt_BR.html. Citado na página 13.

RAUBER, T.; RÜNGER, G. Parallel programming: For multicore and cluster systems. [S.l.]: Springer Science & Business Media, 2013. Citado na página 27.

STALLINGS, W.; FIGUEIREDO, C. C. de; FIGUEIREDO, L. C. de. Arquitetura e organização de computadores: projeto para o desempenho. [S.l.]: Prentice-Hall, 2009. Citado na página 25.

TANENBAUM, A. S. Redes de computadoras. [S.l.]: Pearson educación, 2003. Citado na página 25.

TANENBAUM, A. S. *Modernos, Sistemas Operativos*. [S.l.]: Prentice-Hall. Interamericana SA Madrid, 2009. Citado na página 26.

TANENBAUM, A. S.; STEEN, M. V. Sistema Sistemas Distribuídos-princípios e paradigmas-2º Edição. [S.l.]: São Paulo: Pearson Prentice Hall, 2007. Citado na página 25.

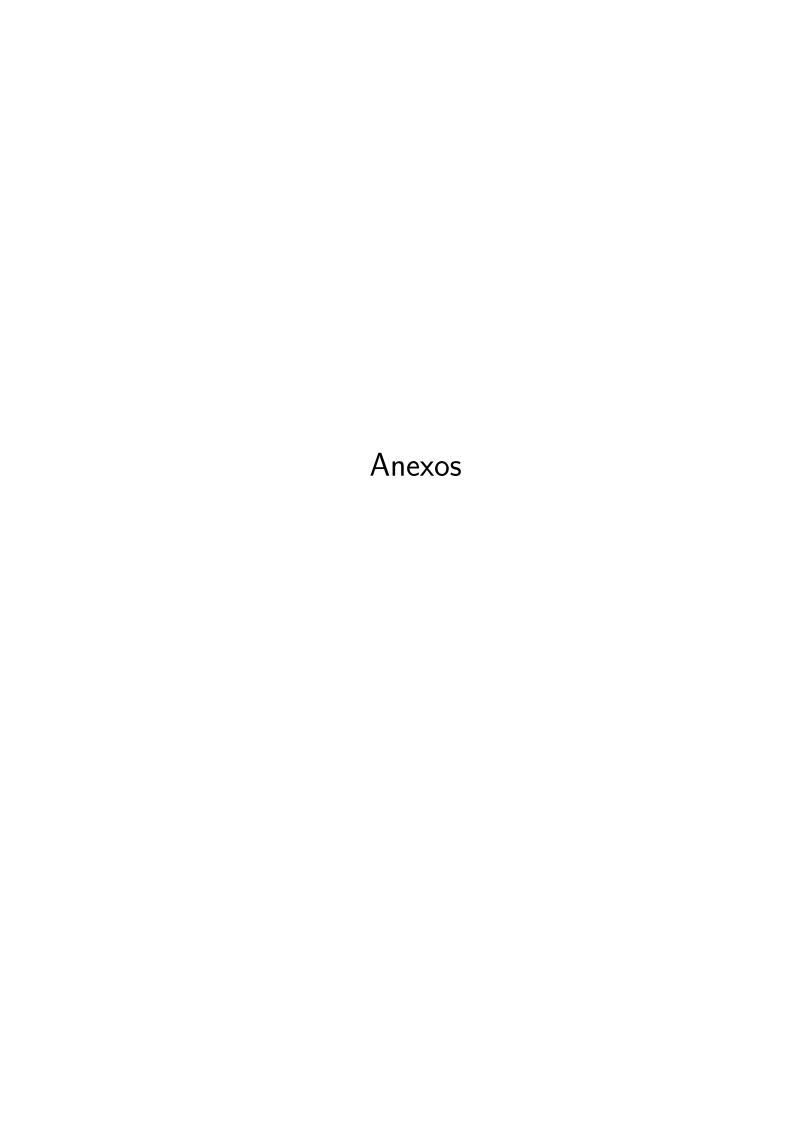
UFABC. Cursos. PROGRAD, 2017. Disponível em: http://prograd.ufabc.edu.br-/cursos. Citado na página 1.

UFABC. *Projeto Pedagógico*. UFABC, 2017. Disponível em: http://www.ufabc.edu.br-/a-ufabc/projeto-pedagogico. Citado 2 vezes nas páginas 1 e 3.

34 Referências

UFABC. Sobre a UFABC. Universidade Federal do ABC, 2017. Disponível em: http://ufabc.edu.br/index.php?option=com_contentview=articleid=16Itemid=3. Citado na página 3.

UFABC, S. de Gestão de P. *UFABC em números*. Pró-Reitoria de Planejamento e Desenvolvimento Institucional UFABC, 2017. Disponível em: http://propladi.ufabc-edu.br/informacoes/ufabc-em-numeros>. Citado na página 1.



ANEXO A - Plano de Atividades

Descrição de atividades típicas do cargo: Executar e/ou acompanhar a manutenção dos sistemas implantados. Operar computadores, suas unidades periféricas e equipamentos de apoio. Transmitir e receber dados. Instruir usuários na utilização de equipamentos e sistemas. Acompanhar e executar a instalação dos equipamentos bem como a manutenção preventiva dos mesmos. Identificar irregularidades nos equipamentos para efeito de manutenção. Copiar arquivos para efeito de segurança. Apoiar serviços de auditoria sobre o uso da rede da Instituição. Executar manutenção em planos de testes em programas/sistemas. Definir e implementar novas soluções tecnológicas. Buscar novas tecnologias em informática para uso na Instituição. Orientar, acompanhar e controlar os recursos computacionais disponíveis, incluindo os serviços desenvolvidos por terceiros. Desenvolver e/ou manter sistemas, recuperar informações em ambientes de Banco de Dados. Coordenar projetos e sistemas aplicativos. Dar suporte às atividades didáticas, de pesquisa e de extensão. Executar outras tarefas de mesma natureza e nível de dificuldade.