

Programação para Administração de Redes de Computadores – Uma Experiência com Estudantes de Computação

Silmar Antonio Buchner de Oliveira^{1,2}, Andréa Pereira Mendonça¹

¹Mestrado Profissional em Ensino Tecnológico. Instituto Federal do Amazonas (IFAM)
Manaus – AM – Brasil.

²Instituto Federal de Rondônia (IFRO). Porto Velho – RO – Brasil.

silmar.oliveira@ifro.edu.br, andrea.mendonca@ifam.edu.br

Abstract. *In this paper we report an experience on conception and execution of a programming course on computer network administration. This course is aimed at students of undergraduate technology courses in Computing related areas. Organized according to a flipped classroom approach, the course focused on solving practical problems by using the Python programming language. The results obtained evidence student's progress in automatising solutions for network equipment and user actions monitoring, log analysis, data synchronizing as well as automatic alerts issuing. The teaching materials and the dynamics of the course will be integrated into a guideline to aid other teachers of computer network affined courses.*

Resumo. *Neste artigo relatamos uma experiência de criação e execução de um curso de programação para administração de redes de computadores, voltado para estudantes de Cursos Superiores de Tecnologia na área de Computação. Implementado no formato de sala de aula invertida, o curso focou na resolução de problemas práticos utilizando a linguagem Python. Os resultados evidenciaram progresso dos estudantes na automatização de soluções para monitoramento de equipamentos de redes e ações de usuários, análise de logs, sincronismos de dados e emissão automática de mensagens de alertas. O material de ensino e dinâmica do curso serão integrados em um guideline para auxiliar outros professores de redes de computadores.*

1. Introdução

Os currículos dos Cursos Superiores em Computação no Brasil, independente da categoria (bacharelado, tecnologia ou licenciatura), possuem em comum a presença de duas disciplinas: (i) **programação** que integra a área de formação básica; e (ii) **redes de computadores** que integra a área de formação tecnológica (BRASIL, 2012).

Tratando especificamente sobre os Cursos Superiores de Tecnologia (CST), observamos que a disciplina de redes de computadores é lecionada priorizando aspectos teóricos, tais como: topologia de redes; protocolos de comunicação, em especial, o modelo de referência OSI (*Open System Interconnection*) e a pilha de protocolos TCP/IP (*Transfer Control Protocol/Internet Protocol*); tipos de redes e aspectos básicos de interconexão de redes (WANG; BLUM; MCCOEY, 2014). Por outro lado, a disciplina de programação é ensinada dentro de uma perspectiva prática, voltada para a resolução de problemas com a adoção de uma linguagem de programação.

Ocorre que, ao ingressarem na vida profissional, os estudantes dos Cursos Superiores de Tecnologia são demandados para a resolução de problemas práticos. No caso de atuação na área de redes de computadores, são comuns demandas voltadas para administração de redes de computadores, as quais envolvem automatização de procedimentos para monitoramento, controle e gerenciamento de serviços, como execução de *backups*, sincronismo de dados, verificação de consumo de recursos, envio de mensagens de alerta via *e-mail*, etc. Estas demandas, por sua vez, não podem ser atendidas com base em uma formação de redes pautada apenas em aspectos teóricos e dissociada dos conhecimentos de programação.

Dada esta lacuna na formação, relatamos neste artigo a proposta e execução de um curso de programação para administração de redes de computadores, voltado para estudantes de CST. O curso focou na resolução de problemas práticos, adotou Python como linguagem de programação, foi administrado na modalidade de ensino híbrido, mais especificamente, no modelo de sala de aula invertida, combinando estudo autônomo e presencial, os quais foram conduzidos por meio de roteiros de aprendizagem (RA).

2. Ensino de redes de computadores nos CSTs

Os Cursos Superiores de Tecnologia (CSTs) possuem regulamentação específica e uma delas é o Catálogo Nacional de Cursos Superiores de Tecnologia, disponibilizado pela Secretaria de Educação Profissional e Tecnológica do Ministério da Educação (SETEC/MEC). Os Catálogos são organizados por eixos, sendo o eixo Informação e Comunicação aquele que integra os cursos relacionados à área de Computação/Informática, como CST em Análise e Desenvolvimento de Sistemas, CST em Sistemas para Internet, CST em Redes de Computadores, CST em Jogos digitais, entre outros (BRASIL, 2016).

Por sua característica direcionada à prática profissional, os CSTs têm duração mais curta (entre 2 e 3 anos), em comparação com os cursos de bacharelado. Seus egressos possuem um perfil profissional dotado de compreensão sobre o processo produtivo, fundado no saber tecnológico e no ato de fazer, com autonomia para tomada de decisões profissionais, indo além do simples domínio operacional de técnicas de trabalho. Geralmente, os CST são oferecidos pelos Institutos Federais de Educação, pertencentes à rede Federal de Educação Profissional, Científica e Tecnológica, criada em 2008. Há também a oferta de CST em universidades públicas e Instituições de Ensino Superior particulares (BRASIL, 2006; BRASIL, 2002).

A fim de caracterizar o perfil das disciplinas de redes de computadores e de programação em CST, realizamos uma pesquisa *ad hoc* em cursos do eixo Informação e Comunicação, oferecidos pelos Institutos Federais de Educação. Como resultado desta pesquisa, identificamos uma relação de 60 (sessenta) cursos, distribuídos em 40 (quarenta) instituições, conforme informações obtidas no sistema e-Mec¹ do Ministério da Educação.

A partir da identificação dos cursos, analisamos as ementas das disciplinas iniciais de redes de computadores e identificamos que as mesmas eram essencialmente teóricas, baseadas em alguma edição das obras de Andrew S. Tanenbaum (Redes de

¹ <http://emec.mec.gov.br>

Computadores) e de James F. Kurose (Redes de Computadores e a Internet), centradas no modelo OSI e na pilha de protocolos TCP/IP.

A qualidade das obras é indiscutível. Contudo, os resultados da pesquisa *ad hoc* apontam um distanciamento entre teoria e prática na forma como a disciplina é ensinada, no tocante à resolução de problemas de redes de computadores, quando considerado o contexto profissional no qual atuam os egressos dos CST.

Na pesquisa, identificamos 9 (nove) CST que possuíam em sua matriz curricular uma disciplina cuja denominação (por exemplo, Programação de Scripts e Programação Aplicada à Gerência de Redes de Computadores) indicava a junção de redes de computadores e programação. Entretanto, não identificamos informações sobre como as mesmas são aplicadas, de forma que se outro professor quiser repetir a experiência, terá dificuldades, pois não encontrará planejamento, metodologias detalhadas ou problemas práticos abordados nestas disciplinas.

Destacamos ainda que, nos CSTs do Eixo Informação e Comunicação há pelo menos uma disciplina de programação no currículo, ensinada no primeiro ano do curso, antes da disciplina de redes de computadores. Tal organização do currículo poderia resultar no aproveitamento dos conhecimentos adquiridos em programação, dentro da disciplina de redes de computadores, possibilitando aos alunos a resolução de problemas práticos com o auxílio de programação.

Corroborando com os resultados de nossa pesquisa *ad hoc*, Chen (2014) aponta que os cursos de redes de computadores possuem muito conteúdo teórico, abstrato e de difícil entendimento, muitas vezes prejudicando o interesse na aprendizagem e resultando em uma capacidade prática deficitária. O autor sugere como solução a adoção de atividades práticas para a resolução de problemas reais. Kurose et al. (2002) também fazem críticas ao ensino essencialmente teórico nos cursos de redes de computadores e citam a importância de utilizar laboratórios com aplicações práticas. Inclusive, levantam perspectivas sobre quando e quanto programar em redes de computadores. Enfatizam ainda que os estudantes de redes de computadores que aprendem a codificar programas podem ter mais facilidade para internalizar e entender conceitos avançados, além de desenvolver um conhecimento “comercializável”. Wang, Blum e McCoey (2014) ao discutirem o ensino de redes de computadores para iniciantes, sugerem que, além da introdução geral e tradicional, devem proporcionar atividades práticas que envolvam programação para resolução de problemas.

Frente a esta lacuna de formação, desenvolvemos a proposta de um curso de programação para administração de redes de computadores, focado na resolução de problemas práticos, que adota a linguagem de programação Python e organiza as práticas de ensino-aprendizagem de acordo com o ensino híbrido, mais especificamente, o modelo de sala de aula invertida, como será detalhado na seção seguinte.

3. Metodologia do curso

O curso está estruturado em dois módulos: o primeiro, abrangendo assuntos referentes ao monitoramento de redes e de ações de usuários; o segundo, integrando assuntos sobre a preservação das informações e comunicação de eventos via e-mail e Telegram. Cada assunto está associado a uma temática, cujo desdobramento se dá pela integração de conteúdos sobre redes, sistemas operacionais e programação, conforme pode ser

observado no Quadro 1. Além disso, para cada assunto, há um conjunto de problemas propostos, nos quais os estudantes devem integrar teoria e prática, propondo soluções automatizadas com uso de programação. Os assuntos abordados em cada temática são cumulativos e suas complexidades evoluem conforme o andamento do curso.

Quadro 1. Módulos, temas, assuntos e conteúdos.

Módulo 1 - Monitoramento de redes de computadores
<p>Tema 01: Identificando a movimentação da vizinhança</p> <p>Assunto: Monitoramento de disponibilidade de equipamentos e serviços em rede.</p> <p>Conteúdos principais: <u>Redes:</u> Comando “ping”; Protocolo SNMP. <u>Programação:</u> Bibliotecas e métodos; variáveis; entrada e saída de dados. <u>S.O. Linux:</u> Estruturas de arquivos e diretórios; - Monitoramento de carga e de estados de componentes.</p> <p>Problemas propostos: Detecções de problemas de lentidão ou perda de acesso, através de monitoramentos via SNMP e “ping”, relacionados ao estado de máquinas e componentes.</p>
<p>Tema 02: Cuidando do Habitat</p> <p>Assunto: Monitoramento de estado de discos e partições, em rede.</p> <p>Conteúdos principais: <u>Redes:</u> Protocolo NFS; Protocolo SNMP. <u>Programação:</u> Bibliotecas e métodos; variáveis; entrada e saída de dados. <u>S.O. Linux:</u> Compartilhamento de dados; Estruturas de arquivos e diretórios;</p> <p>Problemas propostos: Monitoramentos remotos de discos e partições como prevenção de estouros de espaços de armazenamento.</p>
<p>Tema 03: De olho nas intenções</p> <p>Assunto: Monitoramento de ações de usuários.</p> <p>Conteúdos principais: <u>Redes:</u> Protocolo SSH, Acesso remoto; <u>Programação:</u> Bibliotecas e métodos; variáveis; entrada e saída de dados; Operações sobre arquivos; Operações com <i>strings</i> e datas; Laços; Decisão. <u>S.O. Linux:</u> Editores de textos; Acesso remoto; Arquivos de configuração; Hierarquia de usuários; Históricos de ações; Data e hora;</p> <p>Problemas propostos: Monitoramentos de ações de usuários através de verificação automatizada de <i>logs</i> do sistema operacional.</p>
Módulo 2 - Preservação de informações e comunicação de eventos
<p>Tema 04: Construir é difícil. Reconstruir pode ser impossível</p> <p>Assunto: Compactação de dados, cópias de segurança em rede e registros de atividades (<i>logs</i>).</p> <p>Conteúdos principais: <u>Redes:</u> <i>Backup</i>; Compartilhamento de arquivos; Protocolo NTP; Protocolo NFS. <u>Programação:</u> Bibliotecas e métodos; variáveis; entrada e saída de dados; Operações com <i>strings</i> e datas; Laços; Decisão. <u>S.O. Linux:</u> Empacotamento e compactação de dados; Agendamento de tarefas; Sincronismo de data e hora; Arquivos de <i>logs</i>.</p> <p>Problemas propostos: Agendamentos de execuções de <i>backup</i> em rede, com verificação de etapas de execução e gravação em arquivos de <i>log</i>.</p>
<p>Tema 05: Um é pouco. Então tenha dois (ou mais!)</p> <p>Assunto: Sincronismo de dados em rede.</p> <p>Conteúdos principais: <u>Redes:</u> <i>Backup</i>; Políticas de <i>backup</i>; Sincronismo de dados remotos; Compartilhamento de arquivos; Protocolo SSH; Protocolos NFS. <u>Programação:</u> Bibliotecas e métodos; variáveis; entrada e saída de dados; Operações com <i>strings</i> e datas; Decisão. <u>S.O. Linux:</u> Sincronismo de dados; Agendamento de tarefas; Arquivos de <i>logs</i>.</p> <p>Problemas propostos: Agendamentos de execuções de sincronismo de dados em rede, com verificação de etapas de execução e gravação em arquivos de <i>log</i>.</p>
<p>Tema 06: Seja o primeiro a saber</p> <p>Assunto: Automatização de mensagens de eventos e de alertas via <i>e-mail</i> e mensageria instantânea (Telegram).</p> <p>Conteúdos principais: <u>Redes:</u> Serviços de e-mail; Protocolo SMTP; Protocolos SSL/TLS e StartTLS. <u>Programação:</u> Bibliotecas e métodos; variáveis; entrada e saída de dados; Envio de <i>e-mails</i>; Operações com <i>strings</i>. <u>S.O. Linux:</u> Envios de <i>e-mails</i>; Envio de mensageria instantânea; Sincronismo de dados; Agendamento de tarefas; Configuração de serviços.</p> <p>Problemas propostos: Automatizações de procedimentos com verificações de etapas de execuções e, em função dos resultados, envio de avisos e alertas via e-mail e/ou Telegram.</p>

Fonte: Autoria Própria.

Para a definição dos conteúdos, realizamos uma pesquisa em livros nacionais e internacionais que tratam sobre redes de computadores, conforme apresentado no Quadro 2. Com base nesta pesquisa, selecionamos os assuntos mais frequentes e identificamos também os tipos de problemas associados aos conteúdos.

Quadro 2. Relação livro x conteúdo

Legenda dos conteúdos: 1-Monitoramento de Redes; 2-Compartilhamento de dados; 3-Envio de *e-mails*/relatório; 4-Gerenciamento de sistemas de arquivos; 5-Gerenciamento de usuários/acesso; 6-*Login*; 7-Monitoramento de sistemas; 8-Gerenciamento de Arquivos e Diretórios; 9-*Backup*; 10-SSH/acesso remoto; 11-Análise de *logs*; 12-Compressão de dados; 13-Authenticação segura; 14-Expressões regulares; 15-Manipulação de *strings*.

Relação Livro x Conteúdo		Conteúdo														
Livro	Ano	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1 Administração de redes com scripts	2007															
2 Automating Linux and Unix System Administration	2009															
3 Automating System Administration with Perl	2009															
4 Automating Windows Administration	2004															
5 Automating Windows Server 2008 Administration with Windows PowerShell	2008															
6 Core Python Applications Programming	2012															
7 Essential System Administration-Tools and Techniques for Linux and Unix	2002															
8 Foundations of Python Network Programming	2014															
9 Learning Python Network Programming	2015															
10 Linux Command Line and Shell Scripting Bible	2015															
11 Linux System Administration Recipes	2009															
12 Pro Python System Administration	2014															
13 Python for Unix and Linux System Administration	2008															
14 Shell Scripting Recipes	2015															
15 Unix and Linux System Administration Handbook	2011															
16 Wicked Cool Shell Scripts – 101 scripts	2004															
Total de ocorrências		9	8	8	8	8	7	7	7	6	6	4	4	3	3	3

Fonte: Autoria Própria.

A dinâmica para administração dos conteúdos e resolução dos problemas foi organizada obedecendo ao modelo de ensino híbrido de sala de aula invertida, criado por Bergmann e Sams (2012). De acordo com este modelo, uma parte da aprendizagem é feita de forma autônoma pelo estudante, fora da sala de aula, utilizando recursos tecnológicos. A outra parte é realizada em sala de aula, com a presença de um professor. Assim, a carga horária do curso de 60 (sessenta) horas, foi distribuída em 40 (quarenta) horas para estudos presenciais e 20 (vinte) horas para estudos autônomos.

Para guiar o aluno tanto no estudo presencial quanto no autônomo, baseados nos conteúdos das obras elencadas no Quadro 2, definimos os Roteiros de Aprendizagem (RA) que são instrumentos que conduzem o aluno por um itinerário de estudo a fim de que ele atinja os resultados pretendidos da aprendizagem. Para cumprir os assuntos descritos no Quadro 1, doze (12) roteiros foram elaborados, sendo 6 (seis) roteiros de aprendizagem para estudo autônomo² e 6 (seis) roteiros para estudo presencial³. Os RAs para estudo autônomo e presencial se assemelham na sua estrutura geral, a diferença reside na seção de tarefas, conforme pode ser verificado no Quadro 3.

² Um exemplo de RA autônomo (RA Autônomo 01) está disponível em <https://goo.gl/t6wTj5>.

³ Um exemplo de RA presencial (RA Presencial 01) está disponível em <https://goo.gl/qeTGDu>.

Para auxiliar na realização dos estudos autônomos, foram criadas 27 vídeoaulas, com tempo médio de 7 minutos cada, num total de 3 horas e 28 minutos, postadas no Youtube. Também foram utilizados vídeos de terceiros como os que tratavam sobre Python e redes, tais como: Python para Zumbis⁴, Curso em vídeo⁵ e Nic.br⁶.

Quadro 3. Estrutura geral dos Roteiros de Aprendizagem (RAs).

Cabeçalho
Identifica o módulo corrente, o número do roteiro, o tema abordado, o tipo de estudo (presencial ou autônomo) e a carga horária prevista.
Descrição geral
Apresenta orientações sobre como seguir o RA.
Resultados pretendidos da aprendizagem
Descreve os objetivos da aprendizagem a serem atingidos pelos estudantes.
Conteúdos envolvidos
Apresenta os conteúdos abordados, referentes a redes de computadores, programação e sistemas operacionais (Linux).
Tarefas
Nos estudos autônomos , orientam a evolução do estudante sobre os conteúdos preparatórios para a resolução dos problemas que serão apresentados nos estudos presenciais. Tais tarefas são dotadas de questões cujas resoluções são guiadas por vídeos, tutoriais <i>online</i> e <i>links</i> de páginas <i>web</i> para a aplicação prática em um laboratório virtual, criado especificamente para o curso. Os estudos presenciais são compostos de dinâmicas para resolução de problemas, inspiradas em Bender (2014) e em Biggs e Tang (2011), divididas em: <u>Aquecimento</u> - Durante os primeiros 30 minutos de aula, os estudantes expõem suas dúvidas sobre o estudo autônomo, em uma discussão mediada pelo professor. <u>Apresentação do contexto</u> - É apresentado um contexto envolvendo um problema com características reais de redes de computadores. <u>Discussão</u> - São dispostas questões que instiguem o estudante a tentar entender melhor a solução do problema e fomentem uma discussão sobre formas de solucioná-lo. <u>Resolução</u> - O estudante descreve como será a solução para o problema e quais recursos de programação utilizará. Depois constrói um programa em Python consonante com a descrição e o executa no laboratório virtual.

Fonte: Autoria Própria.

O material do curso foi disponibilizado de forma virtual no Google Classroom⁷. Outros canais virtuais também foram utilizados, como os que serviam para elucidação de dúvidas sobre o estudo autônomo, como e-mails e mensagens Whatsapp. De forma complementar, uma vez por semana havia duas horas e meia dedicadas ao atendimento aos alunos, cuja presença era facultativa. Estes atendimentos tinham por objetivo dirimir dúvidas e explicar conceitos, antes de cada estudo presencial. Também foram disponibilizados horários extras no laboratório para que os estudantes pudessem realizar

⁴ <https://www.pycursos.com/python-para-zumbis/>

⁵ <http://www.cursoemvideo.com/course/curso-python-3/>

⁶ <http://www.nic.br/videos/>

⁷ Google Classroom (no Brasil, Google Sala de Aula) é uma plataforma web da Google que reúne um conjunto de serviços (e-mail, drive, gerenciador de tarefas, entre outros) para auxiliar e promover atividades educacionais.

estudos autônomos e sanar dúvidas sobre os estudos presenciais, dispondo de computadores e Internet na própria instituição de ensino.

Três avaliações foram planejadas para o curso: (i) avaliação diagnóstica, realizada antes de iniciar o curso e que tinha por objetivo identificar os conhecimentos prévios dos alunos; (ii) avaliação intermediária, planejada para o final do primeiro módulo; e, (iii) avaliação final, a ser realizada após o término do segundo módulo. Cada avaliação possuía estrutura semelhante aos roteiros de aprendizagem para estudo presencial, com tarefas direcionadas à resolução de problemas de redes com uso de programação.

A primeira experiência de aplicação do curso foi realizada no período de novembro/2017 a janeiro/2018, na forma de um curso de extensão, ofertado no Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, com 30 vagas para estudantes de cursos de nível superior na área de Informática. O pré-requisito era tão somente o estudante ter cursado previamente alguma disciplina de programação. Para este curso, houve 53 (cinquenta e três) inscrições, sendo 30 (trinta) estudantes classificados, conforme o coeficiente de rendimento. Como o período de realização do curso coincidiu com o encerramento do semestre, concorrendo com as avaliações finais dos cursos regulares nos quais os alunos estavam matriculados, e com as férias de final de ano, 15 (quinze) alunos desistiram, mesmo tendo assinando termo de compromisso. Desta forma, os resultados apresentados na próxima seção serão reportados com base nos alunos que concluíram o curso (15 alunos).

4. Resultados

Dos alunos que concluíram o curso, 80% eram do sexo masculino e 20% do sexo feminino. A média de idade do grupo era de 24,2 anos. Identificamos também que 80% dos alunos haviam cursado, no mínimo, uma disciplina de redes na graduação.

Como falamos anteriormente, antes de iniciarmos o curso, os alunos realizaram uma avaliação diagnóstica contendo: (i) questões objetivas sobre a teoria básica de redes de computadores, envolvendo classificação de redes por distribuição geográfica, principais protocolos e modelos de pilhas de protocolos; e, (ii) uma questão prática envolvendo a automatização com verificação de disponibilidade de equipamentos em uma rede, com envio de alerta por e-mail.

Com respeito às questões teóricas objetivas, mais de 84% dos estudantes não acertaram questões básicas como associar o comando *ping*⁵ ao protocolo ICMP (*Internet Control Message Protocol*). Dificuldade semelhante foi percebida quando quase 70% dos estudantes tampouco demonstraram conhecimentos sobre a utilidade do protocolo SNMP⁶ (*Simple Network Management Protocol*). Com respeito à questão prática, 30% dos estudantes deixaram de resolvê-la e, dos que tentaram resolver, seus desempenhos não ultrapassaram 35% de acertos.

Considerando a característica do grupo, no qual 80% dos alunos já haviam cursado uma disciplina de redes de computadores, era esperado melhores resultados nas questões teóricas. Os resultados desta avaliação diagnóstica ratificam as dificuldades de

⁵ O comando *ping* faz parte do protocolo ICMP e é utilizado para testes básicos de comunicação entre computadores.

⁶ SNMP é um protocolo utilizado para gerenciamento de redes de computadores e seus componentes.

aprendizagem em redes, conforme contextualizamos na Seção 2.

Com respeito à realização dos roteiros de aprendizagem administrados ao longo do curso, os estudantes concluíram 86% dos RA para estudo autônomo e 87% dos RA para estudo presencial. Embora a evolução dos estudantes tenha sido evidenciada ao longo do curso, percebemos na maioria dos alunos uma dificuldade inicial em desenvolver seus estudos autônomos, pois a experiência de “estudar sozinho” parecia inédita e/ou desconfortável para muitos. Em virtude disso, houve entregas posteriores ao prazo estabelecido e também ausências de entregas, conforme constatado pelos números apresentados acima. Cabe destacar que, a não entrega dos RA para estudo autônomo acabava impactando na entrega dos RA para estudo presencial, pois faltava aos alunos a fundamentação conceitual necessária para a resolução dos problemas propostos.

No decorrer do curso, os estudantes foram capazes de automatizar soluções para 12 (doze) problemas, distribuídos em 6 (seis) RA para estudo presencial. Estes problemas evoluíram desde situações simples envolvendo monitoramento de conexões de redes com o comando “*ping*”, até automatização de análise remota de *logs*, monitoramento e emissão de alertas sobre a ocupação de partições de discos de máquinas remotas, sincronismo de dados entre servidores, cópias de segurança em rede, além de envio de alertas automáticos via e-mail e Telegram, orientados à criticidade de cada situação. Os alunos também tiveram que montar e administrar seus laboratórios virtuais, utilizando o sistema operacional Linux sobre a plataforma de virtualização Oracle VM Virtualbox, ferramentas até então desconhecidas pela maioria.

No processo de resolução de problemas, identificamos dificuldades dos alunos com respeito à programação, mais especificamente, com a manipulação de *strings*, arquivos e estrutura de laços de repetição. Também identificamos desconhecimento sobre o sistema operacional Linux, mitigados pela intensificação de atendimento a alunos de forma presencial e remota, indicação de vídeos de terceiros e pela produção de uma apostila envolvendo estes assuntos.

A fim de proporcionar um *feedback* contínuo aos estudantes, ao final de cada estudo presencial, os estudantes recebiam de maneira individual, por e-mail ou Whatsapp informações sobre seu desempenho, com indicação de quais aspectos teóricos e/ou práticos deveriam melhorar. Estes *feedbacks* foram muito importantes e refletiram em melhorias no desempenho individual ao longo do curso.

Na avaliação intermediária, ocorrida ao final do primeiro módulo, a média da turma foi 6,8 pontos (em uma escala de 0 a 10), acima da nota mínima para aprovação no curso e também acima da média aceita para aprovação nos CST regulares que os estudantes cursavam, ambos de 6,0 pontos. Na avaliação final, realizada após o término do segundo módulo, a média da turma foi de 8,7 pontos (em uma escala de 0 a 10), representando uma melhora significativa no desempenho do grupo.

Com base nesta aplicação do curso foi possível identificar aspectos a serem melhorados, como por exemplo: (i) ampliar o material de apoio sobre Python no que diz respeito à manipulação de *strings*, arquivos e estrutura de repetição, com exercícios específicos para consolidar a aprendizagem; (ii) incluir conteúdos básicos sobre o sistema operacional Linux, a fim de proporcionar aos alunos melhor familiaridade com seus conceitos e operações; e, (iii) melhorar as vídeoaulas para esclarecer termos e conceitos básicos de redes, que foram mencionados pelos alunos ao longo do curso.

Com respeito ao estudo autônomo, a aplicação deste curso revelou algumas demandas para professores e gestores de Instituições de Ensino Superior (IES). Primeiro, é necessário que as IES disponibilizem infraestrutura adequada para os alunos, permitindo-lhes, por exemplo, acesso aos laboratórios ou outras dependências fora do horário de aula. Para os professores, é necessário estabelecer com a turma uma “cultura de estudo autônomo” e para isso, os alunos precisam ser orientados. No caso deste trabalho, identificamos os roteiros de aprendizagem como instrumentos adequados para orientar os alunos no estudo. Porém, a preparação dos roteiros e dos materiais de suporte demandam um tempo considerável de planejamento. Em contrapartida, possuem a vantagem de poder ser reutilizados em outras ofertas do curso.

Considerando a plataforma web utilizada, o Google Classroom teve alta aceitação entre os alunos, isto porque todos eram usuários de e-mail da Google e podiam facilmente aproveitar os recursos disponibilizados, com destaque para o compartilhamento de materiais, vídeos e submissão das respostas dos alunos aos roteiros de aprendizagem. Contudo, para as comunicações virtuais e trocas de informação, o tempo de resposta era mais efetivo com a adoção do Whatsapp.

5. Considerações Finais

Neste artigo reportamos o planejamento de um curso de programação para administração de redes de computadores, assim como os resultados de sua aplicação com estudantes de Computação. O curso reuniu conteúdos de redes, sistemas operacionais e programação, em uma perspectiva prática, pautada na resolução de problemas e tendo no modelo de sala de aula invertida a dinâmica para condução do processo de ensino-aprendizagem.

O planejamento do curso demandou seis meses de trabalho e incluiu a elaboração da ementa, conteúdos, organização dos roteiros, elaboração dos problemas práticos, criação dos recursos de apoio (vídeos, textos, códigos, laboratórios virtuais, etc.), além de identificação de recursos disponíveis na Internet que pudessem ser reutilizados no curso para auxiliar os alunos no estudo autônomo.

Como trabalhos futuros, desenvolveremos um *guideline* para professores, disponibilizando todo o planejamento e recursos produzidos para o curso. Acreditamos que isso possa contribuir com outros docentes que ministram a disciplina de redes de computadores. Almejamos também desenvolver uma análise focada no nível de complexidade dos problemas propostos, a fim de identificarmos pontos de correção e redimensionamento dos problemas e dos próprios conteúdos associados.

Adicionalmente, realizaremos novos estudos com a aplicação da proposta de ensino-aprendizagem descrita neste artigo, pois compreendemos que a quantidade de estudantes envolvida nesta primeira experiência representa uma limitação do trabalho. Assim, em aplicações futuras, pretendemos oferecer o curso como uma disciplina do currículo, administrada durante um semestre acadêmico, o que permitirá uma avaliação mais criteriosa do desempenho dos estudantes, suas dificuldades, assim como das potencialidade e limitações do próprio curso, a fim de que possamos ampliar as discussões sobre Educação em Computação, mais especificamente neste trabalho, sobre o ensino de redes de computadores.

Referências

- BENDER, W. N. **Aprendizagem baseada em projetos** – Educação diferenciada para o Século XXI. Porto Alegre: Penso, 2014.
- BERGMANN, J; SAMS, A. **Flip YOUR Classroom** – Reach Every Student in Every Class Every Day. Alexandria: ISTE, 2012.
- BIGGS, J.; TANG, C. **Teaching for Quality Learning at University**. 4. ed. Berkshire, England: Society for Research into Higher Education & Open University Press, 2011.
- BRASIL. Ministério da Educação (2002). **Parecer Conselho Nacional de Educação – Conselho Pleno nº 29/2002**. Institui as Diretrizes Curriculares Nacionais Gerais para a organização e o funcionamento dos cursos superiores de tecnologia. Disponível em: <<http://portal.mec.gov.br/cne/arquivos/pdf/cp29.pdf>>. Acesso em: 23 ago. 2016.
- BRASIL. **Parecer CNE/CES No 136/2012**. Institui Diretrizes Curriculares Nacionais para os cursos de graduação em Computação.
- BRASIL. Ministério da Educação (2006). **Decreto n.º 5.773, de 9 de maio de 2006**. Dispõe sobre o exercício das funções de regulação, supervisão e avaliação de instituições de educação superior e cursos superiores de graduação e sequenciais no sistema federal de ensino. Disponível em: <http://www.planalto.gov.br/ccivil_03/ato2004-2006/2006/decreto/d5773.htm>. Acesso em: 01 set. 2016.
- BRASIL. Secretaria de Educação Profissional e Tecnológica / Ministério da Educação (2016). **Catálogo Nacional de Cursos Superiores de Tecnologia**. 3a edição. Disponível em: <http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=44501-cncst-2016-3edc-pdf&category_slug=junho-2016-pdf&Itemid=30192>. Acesso em: 10 fev. 2018.
- CHEN H, Discussion on the Teaching Methods of Applied Undergraduate Computer Network Courses. 3rd International Conference on Science and Social Research (ICSSR 2014), 2014, Paris. **Proceedings...** Paris: Atlantis Press, 2014, p. 127-130.
- KUROSE, J. F.; et al. Workshop on computer networking: curriculum designs and educational challenges. **ACM SIGCOMM Computer Communications Review**, v. 32, n. 5, p. 1–9, Nov. 2002.
- WANG, Y; BLUM, T; McCOEY, M. Teaching a networking class for freshmen: course design and lessons learned. In: Proceedings of the 15th Annual Conference on Information technology education, 2014, Atlanta. **Proceedings...** Atlanta: 2014, p. 9-14.