

Um Relato sobre a Monitoria da Disciplina de Redes de Computadores no Curso de Sistemas e Mídias Digitais

Vitor Rodrigues Costa¹, Windson Viana¹, Emanuel Ferreira Coutinho¹

¹Sistemas e Mídia Digitais - UFC Virtual - Universidade Federal do Ceará (UFC)
Fortaleza – CE – Brasil

vitorc1206@gmail.com, windson@virtual.ufc.br, emanuel@virtual.ufc.br

Abstract. *The Computer Networks course is an elective teaching unity in the Digital Media Systems (SMD) undergraduate course. It has a theoretical and practical nature. During the classes, students apply the theoretical content in practical activities in the laboratories. After the analysis performed in 2015, with network and protocol-oriented tools, we implemented an improvement of the practical activities in the 2016.1 Semester. Some of the used tools were: OpenSSL, Putty, Telnet, SSH, Wireshark, and Cisco Packet Tracer. The modifications were well received by the students, resulting in an increase of participation and high grade in the activities.*

Resumo. *A disciplina de Redes de Computadores é uma disciplina eletiva ofertada no curso Sistemas e Mídias Digitais, sendo de natureza teórica e prática. Ao longo da disciplina, os alunos aplicam o conteúdo visto em aulas teóricas em atividades práticas em laboratórios, e realização de diversos trabalhos práticos. Após análises realizadas no ano de 2015, com ferramentas voltadas ao estudo de redes e protocolos, um aperfeiçoamento das atividades práticas foi implantado no semestre 2016.1. Algumas ferramentas foram utilizadas: OpenSSL, Putty, Telnet, SSH, Wireshark e Cisco Packet Tracer. As modificações das atividades em geral foram bem aceitas pelos alunos, melhorando a sua participação e as notas nas atividades.*

1. Introdução

Estudos e avanços em Tecnologias de Informação e Comunicação atualmente denotam uma nova estratégia para o desenvolvimento de práticas, necessitando de novos recursos tecnológicos para o âmbito educacional [Ferreira et al. 2013]. Essa é também a realidade de disciplinas como a de Redes de Computadores.

Segundo Tanenbaum (2010), Redes de Computadores são conjuntos de computadores autônomos interconectados por uma única tecnologia. Redes de Computadores estabelecem uma forma padrão de interligar computadores para possibilitar o compartilhamento de recursos físicos ou lógicos [Mendes 2015]. Para Kurose (2013), a internet é uma rede de computadores que interconecta milhares de dispositivos de computação em todo o mundo.

O desenvolvimento de atividades práticas em disciplinas de Redes de Computadores é essencial para o desenvolvimento de habilidades técnicas fundamentais dos estudantes. Muitas vezes os alunos não contam com um espaço físico completo que suporte a execução das práticas exigidas pela disciplina [Ferreira et al. 2013]. Em relação ao ensino na área de redes de computadores, Hassan (2003) comenta que dentre os problemas e dificuldades encontrados para o desenvolvimento da disciplina estão:

material didático e pouca disponibilidade de equipamentos adequados para a utilização de ferramentas. Isto ocorre em grande parte devido à velocidade em que os avanços tecnológicos acontecem e ao custo de manutenção de um laboratório experimental. Considerando essas observações, a utilização de simuladores acaba se tornando uma prática para minimizar essas dificuldades.

A disciplina eletiva de Redes de Computadores do curso de graduação de Sistemas e Mídias Digitais (SMD) da Universidade Federal do Ceará (UFC) tem como ementa a introdução às redes de computadores e Internet, estudando as camadas do protocolo TCP/IP, camadas de aplicação, controle de congestionamento e fluxo, topologia de rede, endereçamento e roteamento, redes *Peer-to-Peer* e programação em *sockets*. Sua bibliografia básica utiliza os livros do Kurose [Kurose 2013] e do Tanenbaum [Tanenbaum e Wetherall 2010]. A disciplina oferece tanto créditos teóricos quanto práticos, necessitando então de atividades práticas para cumprir os créditos e também para reforçar o material teórico ensinado. Para esse intuito, exercícios que tentam englobar o conteúdo da disciplina foram criados durante todas as edições da disciplina, que até o ano de 2016 conta com cinco anos de existência no SMD.

Por se tratar de uma disciplina com foco digital, foi necessário procurar por softwares com licença comercial livre para o uso no ensino da disciplina de forma a não infringir direitos autorais que poderiam danificar a reputação da universidade onde a disciplina é ministrada.

Na disciplina, até agora, são utilizados seis diferentes softwares para as atividades, totalizado cerca de oito exercícios com a inclusão dos exercícios de *sockets*, que usam um ambiente de desenvolvimento Java para a sua programação.

Uma vez escolhidos os softwares livres, foram criadas as atividades para serem realizadas tanto durante a aula quanto como exercícios de longo prazo. Na maioria das vezes, esses exercícios são considerados como forma de avaliação na disciplina. Contudo, durante o ano de 2016 se detectou certas anomalias nas atividades. Uma delas não abordava totalmente os assuntos descritos na ementa da disciplina, e outra se mostrou muito complicada devido à complexidade do software usado.

A Pró-Reitoria de Graduação da Universidade Federal do Ceará anualmente divulga um edital para todos os cursos de graduação submetam projetos para Monitoria de Iniciação à Docência, podendo ter bolsistas remunerados e voluntários. Seus objetivos são: (i) contribuir para o processo de formação do estudante de graduação da UFC; (ii) proporcionar a participação do estudante monitor nas atividades docentes; (iii) facilitar a interação entre estudantes e professores nas atividades de ensino, visando à melhoria da aprendizagem; (iv) proporcionar ao monitor uma visão de conjunto da disciplina e das experiências da relação teoria e prática; e (v) envolver o estudante nas atividades de ensino associadas ao planejamento e à pesquisa.

Para minimizar as anomalias detectadas na disciplina, o objetivo desse ano da Bolsa de Iniciação à Docência da disciplina Redes de Computadores foi a modificação, correção e expansão das atividades de forma a melhorar o aproveitamento teórico e prático da disciplina. Neste artigo, se detalhará as atividades realizadas durante o ano de 2016 dessa bolsa.

O restante do artigo está dividido nas seguintes seções além dessa introdução: a Seção 2 contextualiza o curso e as disciplinas da monitoria; a Seção 3 descreve as modificações ocorridas na atividade com sockets; a Seção 4 apresenta as modificações na prática com o software CPT; e por fim na Seção 5 são descritas as conclusões e trabalhos futuros.

2. Contextualização do Curso de Graduação e da Disciplina

O curso de graduação em Sistemas e Mídias Digitais¹ (SMD), da Universidade Federal do Ceará (UFC), objetiva formar profissionais com conhecimentos especializados em duas grandes áreas principais, Sistemas Multimídia e Mídias Digitais, contribuindo para novos perfis de profissionais que possam sustentar o desenvolvimento de mídias digitais e sistemas multimídia, tais como: sistemas *web*, dispositivos móveis, jogos digitais e animações gráficas. Além disso, o curso possui características de interdisciplinaridade, contendo em seu corpo docente professores das áreas de Computação, *Design* Digital, Educação e Comunicação.

Nos três primeiros semestres do curso, todos os alunos cursam o mesmo conjunto de disciplinas obrigatórias. Essas disciplinas têm o objetivo de oferecer, aos graduandos, uma formação teórica e prática para a compreensão dos elementos fundamentais das áreas de Sistemas Multimídia e Mídias Digitais. No quarto e quinto semestre, o aluno pode definir uma área de concentração, a partir da escolha das disciplinas eletivas mais relacionadas com a área de interesse, ou optar por cursar disciplinas de ambas as áreas de concentração. A estrutura curricular do curso foi organizada de forma a conduzir o aluno a cursar pelo menos uma disciplina que não pertença a sua área de concentração escolhida, assim diversificando os saberes e garantindo a integração entre as áreas.

A partir do quinto semestre o aluno pode escolher um itinerário formativo, que é um conjunto de disciplinas optativas que compõem a organização da sua formação em uma área específica, como jogos digitais, animação gráfica ou sistemas *web*. Caso o aluno opte por não escolher um itinerário formativo específico, o mesmo poderá cursar disciplinas optativas de diferentes itinerários, tendo uma formação mais generalista.

A monitoria possui um bolsista remunerado para apoiar duas disciplinas: Redes de Computadores e Sistemas Distribuídos.

A ementa de Redes de Computadores consta de: introdução às redes de computadores e seu impacto na sociedade; conceitos básicos de redes de computadores; topologias de redes; camada de aplicação: transmissão de arquivos, DNS, SMTP e web; programação em rede; componentes lógicos e físicos em redes; ligações inter redes; conceitos básicos em transmissão de dados; introdução às redes sem fio; e redes de computadores e jogos digitais.

A ementa de Sistemas Distribuídos é composta por: introdução aos sistemas distribuídos; comunicação e sincronização em sistemas distribuídos; comunicação inter processos utilizando RPC/RMI; arquiteturas e tecnologias para o desenvolvimento de aplicações distribuídas; conceitos e criação de plataformas de middleware; e arquiteturas orientadas a serviço.

¹ Sistemas e Mídias Digitais (SMD) - <http://smd.virtual.ufc.br/>

3. Modificações na Atividade de *Sockets*

Das atividades que foram modificadas de alguma forma esse ano, destaca-se a atividade de *sockets*, com o intuito de ensinar os alunos a programarem *sockets* usando a linguagem de programação Java.

Um soquete de rede (*socket*) é um ponto final de um fluxo de comunicação entre processos através de uma rede de computadores, ou seja, um receptor final. *Sockets* fornecem um ponto de comunicação entre processos. Qualquer mensagem enviada de um processo para outro deve passar por uma rede subjacente, e esse processo envia mensagens para a rede, recebendo mensagens dela por meio de uma interface de software denominada *socket* [Kurose 2013].

A atividade demonstra a comunicação entre cliente e servidor usando um pequeno exemplo de jogo de corrida, em que o servidor teria a interface enquanto os clientes enviariam as informações necessárias para se inscreverem na corrida por meio de comunicação *socket*.

Foi detectado a necessidade de mudança por diversos fatores. Um deles foi a limitação de conexões que o servidor aceitava. Na sua versão inicial, apenas quatro alunos podiam se inscrever na corrida simultaneamente. Isto se provou improdutivo quando se compara com o número de estudantes que se inscreveram na disciplina. Outro motivo, foi relacionado à programação. A atividade foi separada em quatro níveis de dificuldade para que os alunos pudessem tentar comunicações de modos mais difíceis e complexos, mais próximos da vida real. Isso criou quatro programas diferentes que praticamente reusavam o mesmo código com pouquíssimas modificações.

Um dos maiores fatores para modificar a atividade foi que a ementa cita a prática de comunicação por *socket* tanto em TCP (*Transmission Control Protocol*) quanto em UDP (*User Datagram Protocol*). Entretanto, a atividade apenas demonstrava TCP. Por fim, a própria prática já não parecia tão interessante como foi relatado pelo próprio professor da disciplina. A Figura 1 exibe a tela da antiga aplicação de *socket*.

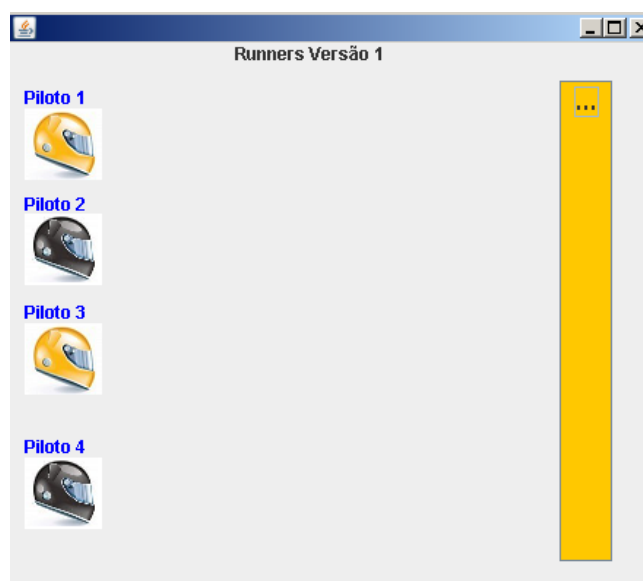


Figura 1. Atividade de *socket* antiga

Para modificar esse quadro, a atividade foi ajustada para que se tornasse um jogo, pois tanto geraria interesse dos alunos quanto se encaixaria na formação de um aluno do curso. Com esse fim, as mecânicas da atividade foram alteradas. Uma dessas foi a adição de um gerador de números aleatórios que determina a velocidade de cada competidor num certo período de tempo. A imprevisibilidade do resultado final é integral a definição de jogo do grande teórico da área Roger Caillois [Caillois 1957].

Em um lado mais técnico, o programa foi construído com a plataforma de desenvolvimento Eclipse² usando a linguagem de programação Java³. O software foi dividido em duas partes, uma que preocupa-se com o visual do jogo, que usou da biblioteca JPanel, e outra que é responsável por gerenciar as conexões socket do servidor.

A parte visual é encarregada de gerir os eventos que são engatilhados com interações com o usuário, como um clique do mouse ou digitar com o teclado. Entre suas funções está um menu principal que permite o usuário escolher qual tipo de atividade será realizada naquela instância do programa.

Como um dos problemas citados era a falta de integração entre as atividades em um único software, o menu tem como objetivo permitir a seleção dos vários tipos diferentes de protocolos de comunicação que a atividade permite trabalhar.

Protocolos de comunicação definem a forma como o servidor e os clientes se comunicam entre si, e que tipo de mensagens eles esperam receber um do outro. Por exemplo, caso o usuário envie sua identificação (*id*) e senha em uma única linha separados por uma vírgula o servidor tem que estar programado para entender esse tipo de mensagem. Contudo se ele tivesse esperando o *id* e senha separadamente, em duas linhas diferentes, ou talvez separados por um parêntese, ele não entenderia a mensagem do cliente, mesmo que as informações enviadas sejam as mesmas.

Portanto, o propósito de ensinar um protocolo de comunicação é mostrar aos alunos como eles podem padronizar a comunicação entre cliente e servidor. Também é objetivo tratar de problemas como mensagens corrompidas e aplicativos de terceiros tentando se comunicar com o servidor sem autorização, usando um protocolo que não combina com o estabelecido.

O menu mostrado na Figura 2 consiste de cinco botões que determinam qual dos protocolos o servidor vai esperar dos clientes, um campo de *input* que especifica qual porta o *socket* vai ser aberta. Caso nada seja escrito, ele vai assumir que a porta oitenta foi selecionada, e uma breve descrição de como é o protocolo de comunicação daquela atividade é exibida.

A especificação de qual porta foi um pedido do professor que ministra a disciplina para que ele possa usar mais de uma instância do programa ao mesmo tempo, já que para estabelecer um servidor socket é necessário que a porta esteja livre para uso, então dois servidores *socket* têm que usar duas portas diferentes para rodarem simultaneamente.

Os protocolos são separados em níveis de dificuldade em ordem crescente: básico, médio, avançado e *hell*. Também contendo um nível específico para o uso de *sockets* UDP.

² Eclipse - The Eclipse Foundation open source community website - <https://eclipse.org/>

³ Java - <https://www.java.com/>

Os que usam TCP variam entre o uso de vírgulas, palavras específicas para a comunicação e tipos para a troca de informações. Enquanto o nível UDP, devido ao irregular e pouco confiável tipo de conexão que ele possui, exige o uso de identificadores para cada cliente. O objetivo é que o servidor UDP possa saber para quem deve enviar certa mensagem de forma a não quebrar o protocolo de comunicação de outros clientes.

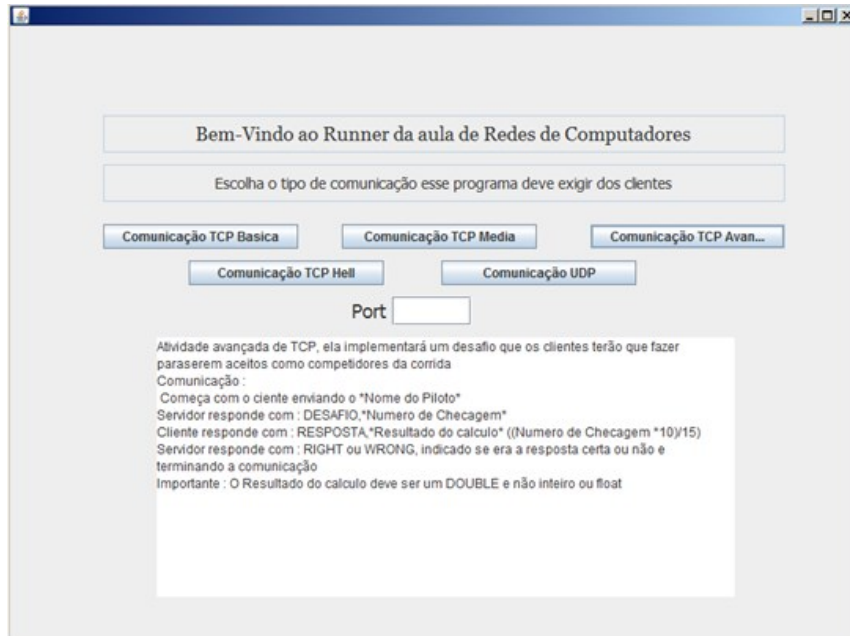


Figura 2. Menu da nova atividade

Finalmente, ao selecionar qual protocolo de comunicação será usado e qual porta o servidor usará, o usuário é levado para a tela mostrada na Figura 3. Nela aparecem várias caixas de texto que irão conter o nome dos jôqueis escolhidos pelos clientes, um botão de largada que servirá para começar a corrida e o visual de um hipódromo para melhor imergir os estudantes no jogo.



Figura 3. Tela da corrida de cavalos do programa

Ao clicar em largada a corrida começara com os estudantes que conseguiram fazer a conexão, e, no caso do protocolo de comunicação avançado e *hell*, também é necessário que eles acertem o desafio que o servidor vai mandar para eles. Em caso de erro, o jóquei é retirado da corrida.

O visual também é encarregado de informar em tempo real o estado da corrida, colocando, junto ao nome do jóquei, qual posição ele está com relação aos outros competidores como mostrado na Figura 3. Também foi associada uma cor aos três primeiros colocados como se estivesse representando um pódio, ouro, prata e bronze.

A corrida termina quando todos os jóqueis chegarem na ponta direita do hipódromo, então um pódio é mostrado junto a um botão. Este permite a volta para o menu principal, para então que outra corrida possa ser preparada. A tela descrita é mostrada na Figura 4.

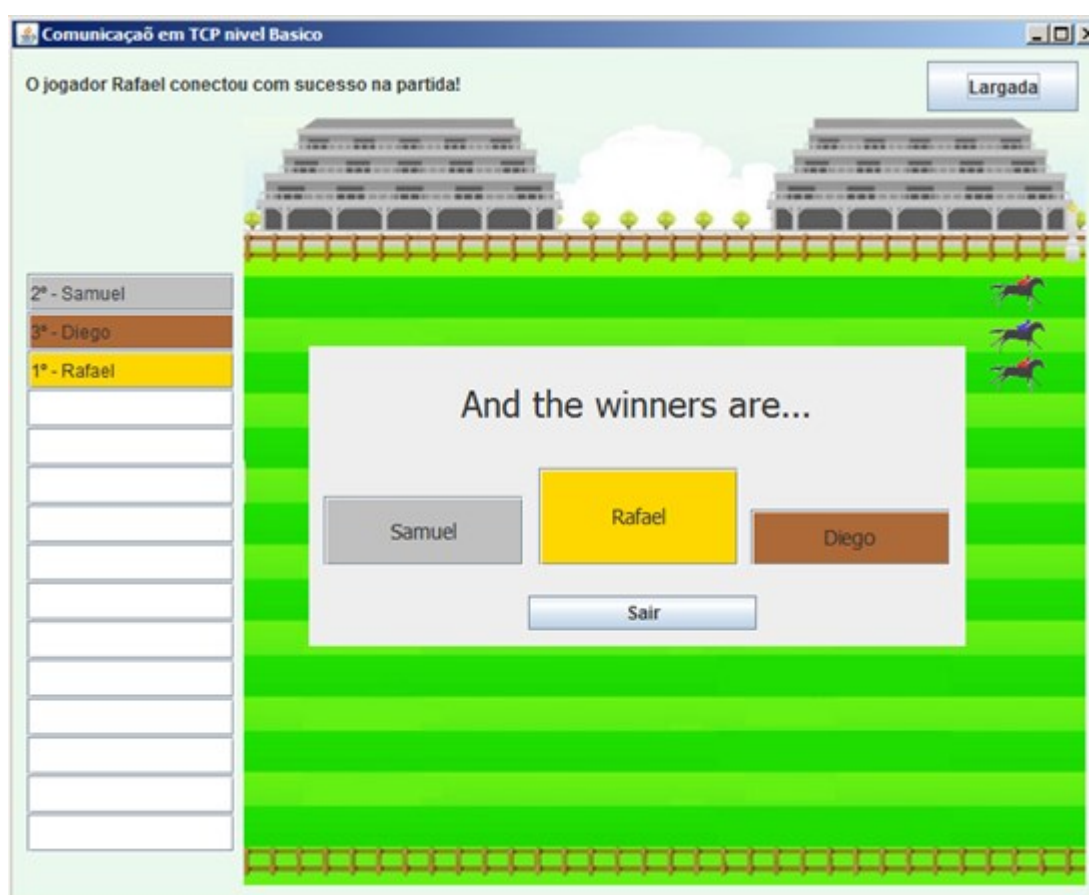


Figura 4. Pódio ao final da corrida

Agora a segunda parte, a parte do servidor é composta por várias classes, entre elas uma para o objeto jóquei, que vai conter as informações que o servidor precisa saber daquele usuário, como o nome, para ser usado no visual da corrida, o endereço IP (*Internet Protocol*), no caso de protocolos de comunicação mais altos vai também ter um número de checagem para o desafio e na conexão UDP haverá o identificador do jóquei.

Na atividade mais básica, os alunos devem fazer um cliente TCP que envia ao servidor apenas o nome do jóquei, o servidor, que estará apenas esperando uma *string*

simples irá usar o que receber como o nome, então mesmo se tiver caracteres especiais ou de separação, o servidor vai supor que eles fazem parte do nome.

O segundo protocolo de comunicação, o de dificuldade média, pedirá os alunos que enviem uma *string* simples, contudo dessa vez é necessário enviar um número de zero a dez e o nome do jôquei. Os dois tem que ser separados por vírgula.

Isso é para simular a checagem que um servidor faz com as informações do cliente, pois caso o cliente não enviasse esse número de checagem ou ele fosse maior que dez ou menor que zero, o servidor detectaria que teria algo de errado com aquele cliente. Assim, ele poderia lhe barrar o acesso, esse seria um exemplo bem arcaico de segurança de conexão e checagem de clientes.

O terceiro trataria de uma atividade mais complicada, a de dificuldade avançada, que para autorizar o cliente a entrar na corrida enviaria um número criado aleatoriamente. Isso também introduz os alunos à recepção de mensagens de um servidor usando um protocolo específico para o cliente. Após receber o número, o cliente tem que fazer um cálculo específico e retornar o resultado deste para o servidor, caso o resultado esteja de acordo com o que o servidor previu ele vai aceitar a conexão com o cliente, caso não ela será rejeitada, esse já seria um exemplo mais avançado de checagem de cliente *socket*.

A última atividade de conexões TCP é a de dificuldade *hell*. Ela, ao invés de enviar *strings* simples, enviará objetos inteiros. Isso é possível graças a Serialização que a linguagem de programação Java dispõe e que permite o envio de objetos entre aplicações que a usam. Para tal, tanto o cliente quanto o servidor devem saber como esse objeto é composto, suas variáveis, funções e interfaces, para poderem então transmitir a informação corretamente.

Com esse intuito, foi implementada a classe *Challenge*. Ela possui três variáveis, uma *string* para o nome do jôquei, um número para a checagem, que vai exigir um cálculo do cliente como na atividade avançada, e uma *string* para mensagens do servidor. A comunicação entre cliente e servidor se dá ao preencher essas variáveis com informações e então enviar o objeto inteiro para o destinatário para que ele possa ler, processar e então colocar mais informações e enviar o objeto de volta.

A atividade de UDP é parecida com a de TCP avançada, contudo tem uma mudança decisiva, devido ao modo que o UDP age. Ao invés de estabelecer conexões, o UDP apenas envia pacotes, se o destinatário recebe ou não o UDP não checará, pois, seu objetivo é mandar o máximo de informação possível, mesmo que ela se corrompa durante o processo. Por isso, que o UDP é dito ser um protocolo não confiável.

O servidor UDP irá distribuir identificadores a todos que requisitarem acesso, para assim o servidor poder arquivar os nomes e endereços IP dos jôqueis com mais facilidade. Ao enviar o identificador é esperado que o cliente mande suas mensagens usando-o, caso não, o servidor irá apenas pensar que é outra solicitação de conexão.

A atividade então segue o método da atividade TCP avançada, com o servidor enviando um número de checagem aleatório e esperando o resultado do cálculo do cliente. Caso esteja certo, o servidor permitirá a entrada do competidor, caso contrário ele é retirado da corrida e da lista de identificadores salvos.

Essas foram as modificações feitas na atividade de *sockets*, as mudanças foram bem recebidas como alega o monitor da disciplina durante as aulas práticas, em que ele

esteve presente, se notou que a parte da corrida deixou os alunos bem focados com a atividade que era o intuito ao tentá-la fazer parecer um jogo.

Também se notou certa dificuldade com as instruções do protocolo de comunicação, pois ele precisa ser feito em sintonia e precisão tanto pelo cliente quanto pelo servidor. Contudo, esse problema pode rapidamente ser resolvido com uma revisão da descrição da atividade que é entregue aos alunos.

Um contratempo da atividade foi notado quando todos os quinze jôqueis possíveis estavam no jogo, o programa ficou extremamente lento e consumia altos índices de memória do computador, isso se deve principalmente pelo uso de threads no programa para gerir a animação dos jôqueis. Assim, quando todos estivessem na corrida haveria quinze threads em execução, e ainda a *thread* do programa principal, rodando ao mesmo tempo, o que geraria atrasos e crashes em computadores menos potentes.

Uma solução do problema exposto acima foi a unificação do *thread* de animação em uma só. Ao invés de quinze threads, foi possível reduzir esse número para uma única que faz quinze iterações nos jôqueis da corrida, isso melhorou o desempenho do programa.

Outras mudanças feitas incluem: tornar a interface mais visualmente agradável e com instruções fáceis de entender; aumento do número de possíveis competidores da corrida para o número de quinze, permitindo a formação de duplas entre os alunos e que todos participem da tarefa simultaneamente, mesmo que a sala esteja cheia; e adição de uma nova atividade que abrange a programação de *sockets* UDP do Java.

4. Modificações na Atividade com o CPT

Outra das atividades modificadas foi a prática utilizando o software CPT⁴ (*Cisco Packet Tracer*), um poderoso simulador de redes de computadores. Ele permite a criação de redes de computadores desde seu hardware o software, incluindo cabos, roteadores, conexões sem fio, simulador de navegadores, acesso ao *prompt* de comando de um dos computadores simulados e vários outros.

Esta atividade possibilitava aos alunos uma rede previamente feita no CPT, a partir dela eles deveriam informar como se dava a comunicação entre os pontos da rede, e caso achassem algum defeito ou anomalia que a concertassem usando o conhecimento aprendido na disciplina.

A prática envolvia a unificação de todos os conhecimentos da ementa da disciplina e se provou uma boa ferramenta de ensino. Entretanto, devido a sua complexidade também se fez necessário uma segunda atividade, menor e que não valeria para a avaliação da disciplina para introduzir os alunos a ferramenta e ao mesmo tempo lhes demonstrar o funcionamento no simulador da diferença entre os protocolos TCP e UDP.

Essa atividade introdutória seria a modificação feita nessa última instância da bolsa. Ela consistia em conectar três pontos de rede, que pode ser interpretado como computadores, laptops, celulares ou servidores, tudo que depende da rede para se comunicar, em um roteador e então programar um dos pontos da rede para ser um servidor, e os outros dois seriam clientes desse servidor.

⁴ Cisco Packet Tracer - <http://www.cisco.com>

O servidor teria duas funções: um servidor *Domain Name System* (DNS) e um servidor *HyperText Markup Language* (HTML). E cada um dos clientes tentaria usar dessas funções do servidor.

O objetivo dessa atividade era a ambientação dos alunos com a simulação de pacotes que o Cisco permite. A ferramenta simula cada movimento de pacotes sendo enviados ou recebidos entre os pontos da rede. Uma observação do professor da disciplina é que seria muito complicado ao vê-lo pela primeira vez sem uma introdução de como ele funciona.

Na atividade, os alunos teriam que visualizar a composição dos pacotes enquanto eles navegavam pela rede simulada, detectando as mudanças e explicando o porquê delas, isso permitiu testar os conhecimentos que eles tinham das camadas de rede que compõe o pacote. Esse exercício foi baseado numa das tarefas que a Cisco fornece, além disso os próprios professores que o usam também formulam exercícios, isso significa uma constante produção de atividades que podem ser aproveitadas na disciplina de Redes de Computadores, o que faz do Cisco uma grande ferramenta a ser pesquisada no futuro da disciplina.

O exercício foi bem aceito e a maioria dos alunos conseguiu terminá-lo. O monitor estava presente durante a atividade e notou que mesmo a descrição da atividade ter sido muito detalhada, incluindo até fotos de cada passo a ser tomado, a ferramenta ainda se mostrou complicada com várias perguntas a partir dos alunos de como tal parte da ferramenta funcionava. Notou-se também que a atividade se mostrou bem curta, praticamente sendo completada em quarenta minutos pelos alunos mais rápidos e uma hora para os mais lentos, isso gerou uma solicitação do professor da disciplina que pediu um aumento da duração da atividade ou talvez adição de parte da atividade de Cisco que valia nota com essa parte introdutória.

As Figuras 5 e 6 exibem telas exemplo do CPT. A própria Cisco, produtora e fornecedora do CPT, fornece exercícios para serem praticados usando seu simulador possibilitando mais oportunidades de atividades no futuro.

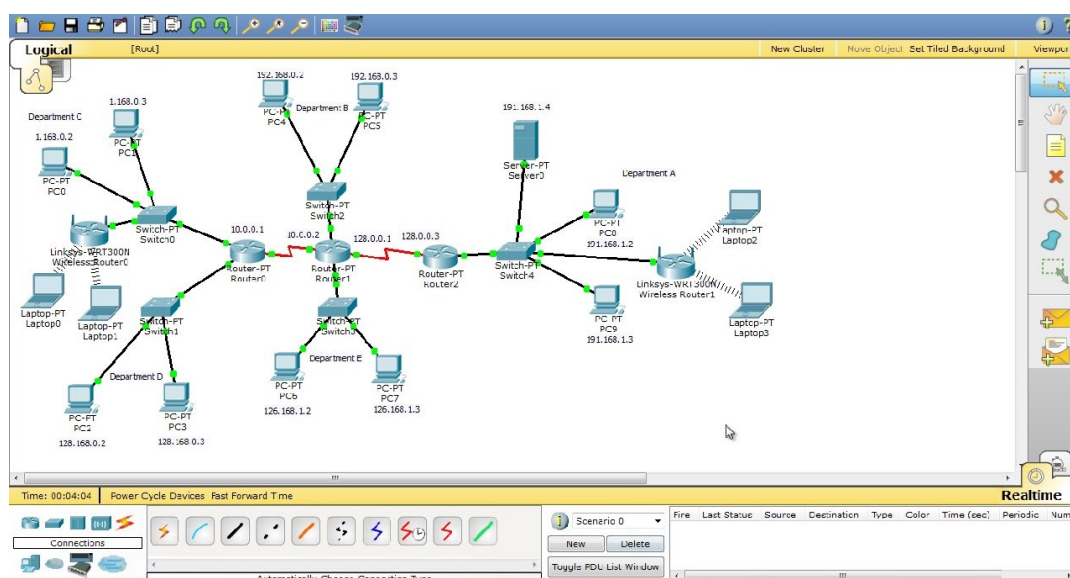


Figura 5. Rede de computadores construída usando o CPT

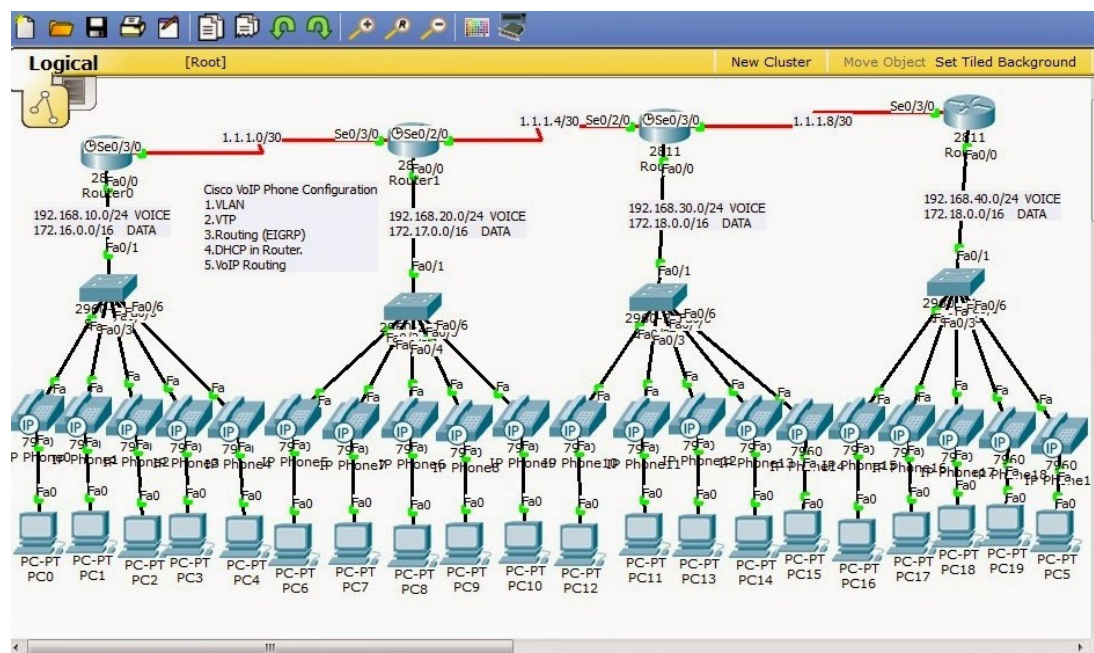


Figura 6. Rede de computadores construída usando o CPT

5. Conclusão

Este trabalho descreveu parte das atividades de monitoria da disciplina de Redes de Computadores do curso de graduação Sistemas e Mídias Digitais. As atividades tiveram boa recepção durante a disciplina pelos alunos, na forma na qual os assuntos teóricos eram abordados na prática e como isso possibilitou uma melhoria no aprendizado deles.

As atividades também contaram com a participação dos alunos, pois a maioria realizou as atividades, e também obtiveram uma boa média nas notas das práticas. Ainda se observa, contudo, a falta de uma atividade que demonstre as conexões *Peer-to-peer*, que pode se tornar um trabalho futuro deste projeto. Além disso, é necessária uma avaliação da efetividade das práticas no real aprendizado dos alunos por meio de uma estratégia mais formal, e que possibilite comparações com turmas diferentes da disciplina.

Finalmente, deve-se notar que as tecnologias sempre mudam e as atividades da disciplina devem se modificar de acordo, sempre necessitando a pesquisa de novos softwares para demonstrar os assuntos aplicados na mesma.

Referências

- Callois, R. (1957) “Les jeux et les hommes”, 9782070326723, Gallimard - Fra.
- Ferreira, K. H. A., de Lima, R. W., Chaves, J. O. M., de Lima, M. V. A. (2013) "Inserindo um Laboratório Virtual para o Ensino de Redes de Computadores", ICBL2013 – International Conference on Interactive Computer aided Blended Learning, Florianópolis.
- Hassan, E. B. (2003) “Laboratório Virtual 3D para ensino de Redes de Computadores”, XIV Simpósio Brasileiro de Informática na Educação, Rio de Janeiro, 2003.

- Kurose, J. F. (2013) “Redes de Computadores e a Internet. Uma Abordagem Top Down”, 6ª. Edição, 9788581436777, Pearson Education - Br.
- Mendes, D. R. (2015) "Redes de computadores - Teoria e prática", 9788575223680, Editora Novatec.
- Tanenbaum, A. S., Wetherall. D. J. (2010) “Redes de Computadores”, 5ª. Edição, 9788576059240, Pearson Education - Br.