

MI Concorrência e Conectividade 2015.2

Segundo Problema: Roletrando Multiplayer

Allen Hichard, Daniel Andrade, Khaick Brito

¹Prof^a Elisângela Oliveira Carneiro
Universidade Estadual de Feira de Santana – UEFS

{allenhichard21, dcndrd, khaickbrito}@gmail.com

Abstract. *This paper has the objective to introduce the development process of a virtual game called Roletrando. This game is similar to the Roda a Roda game, created by Silvio Santos. The created computer game has the 2 and 3 multi-player mode feature. The game shares the screen information like the score and status of each player between the players. The focus of this article is explain the software related concurrence and connectivity concepts, being them the transport protocols, the multicast application in the given problem and the Peer to Peer communication model. Also, we present the followed steps, discussions about the features and project decisions and system failures like the absence of the ranking.*

Resumo. *Este artigo tem como objetivo apresentar o desenvolvimento de um jogo virtual de nome Roletrando, o qual funciona de modo semelhante ao Roda a Roda, criado pelo apresentador Silvio Santos. O jogo criado para computadores possui a característica de permitir a ligação de 2 ou até 3 clientes entre si para formar duplas ou trios iniciando assim a adivinhação de palavras, possibilitando também o compartilhamento de informações entre eles, as pontuações de cada um e o status atual do jogo exibidos na interface. O enfoque desse artigo encontra-se na explanação dos conceitos de concorrência e conectividade associados ao desenvolvimento do software, sendo eles os protocolos de transporte UDP e TCP, a aplicação do Multicast, o que possibilita a troca de mensagens entre os jogadores e também o conceito do tipo de comunicação Peer to Peer. Também serão apresentados os passos adotados para a implementação do programa junto as discussões acontecidas e algumas falhas do sistema, tais como a deficiência do Ranking de melhores resultados.*

Como foi desenvolvido o software?

1. Introdução

A contemporaneidade está sendo marcada por grandes revoluções, do campo social, através da queda de governos ditatoriais e maior participação popular na formação ou reformulação da sua cultura, ao tecnológico, mediante novas descobertas, pesquisas mais bem elaboradas e tecnologias cada vez mais bem aplicadas. Cada situação citada tem um ponto em comum e que serve como base para tais feitos: a comunicação, em especial a intercomputadores, a qual, através da troca de informações e recursos, colaborou com a possibilidade de se alcançar patamares no avanço global jamais atingidos.

Em virtude dessa evolução, a empresa SBT decidiu por reformular e atualizar o jogo Roletrando, visando inseri-lo nas correntes de avanço impulsionadas pelas redes de

computadores. A nova proposta foi a adição de novos modos de jogo agora não mais associando jogadores unicamente ao servidor, mas também a outros jogadores.

Está vago...

A comunicação estabelecida na versão antiga do jogo, por conta da necessidade de conectar os clientes apenas ao servidor central, foi a conhecida como Cliente-Servidor, porém, devido a necessidade da comunicação direta entre os jogadores, a abordagem Peer-to-peer, a qual será conceituada na seção 3.1, foi escolhida para embasar a codificação do programa. Para possibilitar a transmissão das mensagens entre os clientes, se deu a aplicação da comunicação Multicast, tal conceito será explanado na seção 3.3, feita utilizando o protocolo de transporte UDP em virtude da versatilidade e da fácil associação ao Multicast.

A introdução permite ao leitor uma visão geral e completa do estudo feito. Nesta introdução faltam alguns elementos que permita esta visão geral.

2. Fundamentação Teórica

2.1. Protocolo TCP

O Transmission Control Protocol, ou TCP, foi projetado especificamente para oferecer um fluxo de bytes fim a fim confiável numa inter-rede não confiável [Tanenbaum 2003]. Ao contrário do UDP, o TCP ~~tem~~ dá garantia de envio dos pacotes, possui ferramentas de controle de fluxo de pacotes e erros, além de outras ferramentas como retransmissão de pacotes perdidos. No contexto dos protocolos de transporte, mais segurança e confiabilidade significa mais consumo de recursos, ou seja, o protocolo TCP consome mais banda, memória e processamento das entidades envolvidas do que o UDP. O que é UDP?

2.2. Multicasting

Multicasting é a forma de entrega de informações para múltiplos destinatários simultaneamente. Cada usuário em uma rede pode se conectar a um determinado grupo multicasting, e compartilhar ~~de~~ toda informação emitida pelo mesmo. O UDP foi o protocolo de transporte utilizado para fazer a comunicação entre o grupo multicasting e seus usuários, porque o UDP oferece suporte para o tráfego de dados sendo flexível no envio e no recebimento de dados pela rede. O multicast oferece uma forma eficiente de oferecer suporte a aplicativos um-para-muitos de grande largura de banda em uma rede [Rosenzweig et al. 1998]:

- O multicast pode reduzir drasticamente o tráfego da rede ao enviar uma única cópia dos dados.
- Os hosts podem ser configurados para multicast sem a necessidade de atualização de hardware.
- Como os roteadores mais novos já oferecem suporte ao encaminhamento de multicast e aos protocolos de roteamento de multicast, habilitar o multicast em uma rede será prático e econômico.

2.3. Peer to Peer

Conforme [Rocha et al. 2004], a arquitetura Peer to Peer sempre existiu, mas não era conhecido. “A USENET surgiu em 1979, e foi desenvolvida por estudantes universitários dos EUA. Ela permitia que dois computadores trocassem informação, nos primórdios, antes da conexão oferecida pela Internet.” [Rocha et al. 2004]. Nessa aplicação não existia um servidor central e a distribuição de informações e replicação eram feitas por cada usuário.

O Napster e o ICQ foram serviços mundialmente conhecidos e referências do uso da arquitetura P2P em seus sistemas. O ICQ a utilizava para fazer uma comunicação direta entre usuários, substituindo o correio eletrônico, já o Napster, possibilitou o compartilhamento de arquivos em mp3. Ambos, apesar da aplicação do Peer to Peer, utilizavam de um servidor central para armazenar informações de usuários, arquivos e efetuar redirecionamentos. [Referência?](#)

Em acordo com [Rocha et al. 2004], a nomenclatura Peer to Peer tem como tradução do inglês, o significado Par a Par. Tal nome refere-se ao modo como os nós de uma rede de computadores se dispõem, todos se conhecem e se conectam de modo a formar uma cadeia descentralizada. [Rocha et al. 2004] traz a lógica P2P através do seguinte conceito: “Todos os usuários são clientes e servidores, funcionando, assim, de forma totalmente independente e livre da existência de um servidor central”.

A busca através das redes Peer to Peer ~~podem~~ ser feitas por mecanismos conhecidos como: Transmissão de inundação de consultas, a qual consiste em inundar os vizinhos nós com solicitações as quais serão repassadas visando encontrar o almejado, Sistemas de seleção de encaminhamento, “este mecanismo envia a consulta para todos os pares e encaminha seletivamente a consulta para específicos pares que são considerados possíveis locais onde o recurso pode ser encontrado” [Rocha et al. 2004]; Índices centralizados e repositórios, o mecanismo armazena os índices dos pares e dos recursos num servidor central e depois os busca, tal modo pode ser associado ao Napster.

Segundo [Rocha et al. 2004], uma vantagem importante desse modo de arquitetura de rede, a descentralizada, é o fato de que ela possui uma forte interação, sendo mais difícil de ser interrompida devido toda a conexão estar espalhada entre todos os peers. As desvantagens do P2P, por [Rocha et al. 2004], circundam os modos de busca nesse tipo de rede, pois, devido a possibilidade de se criar uma rede muito grande e longa, a busca por alguma informação ou arquivo poderia falhar devido a distância ou se tornar muito lenta, e também questões de segurança como a possibilidade de contaminação por vírus, malware e worms e invasões através de vulnerabilidade nas aplicações de redes P2P.

2.4. Protocolo UDP [Por que aqui?!](#)

O protocolo UDP, que significa User Datagram Protocol, trata-se de um dos protocolos referentes a camada de transporte pertencentes as camadas da comunicação em redes de computadores. Por [Forouzan 2009], o UDP é caracterizado por ser mais simples que o TCP em virtude de sua flexibilidade, pois, enquanto o TCP dedica-se a conexão e a chegada dos dados ao destino, o UDP presa apenas pela velocidade de propagação. Tal protocolo não verifica recebimentos de dados, não possui uma opção de reenvio, não ordena mensagens, não verifica integridade de informações transitadas e também não possui controle de fluxo.

Por [Forouzan 2009], diante de tantas desvantagens perante o protocolo TCP, o qual efetua inúmeras medidas de segurança e preservação de dados para montar uma boa comunicação, o UDP tem como característica a sua velocidade de envio, propagação e recepção, pois possui menos quesitos a serem analisados. Tal característica torna-se essencial para aplicar em situações que prezem pela transmissão em tempo real, como vídeo conferências, streams e jogos.

O datagrama UDP, nome do pacote com os dados, segundo [Forouzan 2009], pos-

sui os campos porta de origem e porta de destino, os quais especificam as portas que serão utilizadas na comunicação. Já o campo tamanho descreve quantos bytes terá o pacote completo. O campo checksum é opcional e faz uma soma verificadora para garantir que os dados estarão livres de erros. Alguns protocolos conhecidos que utilizam o UDP são TFTP, SNMP, DHCP e DNS.

3. Metodologia

Seguindo a metodologia do *Problem Based Learning*¹, após recebido o problema foi iniciado uma discussão envolvendo possíveis soluções para a situação proposta.

Ao contrário do problema anterior, o segundo problema do MI - Concorrência e Conectividade não trouxe uma proposta totalmente nova. Nesse problema, o primeiro produto deveria ser modificado para que suportasse o modo multiplayer na versão de 2 e 3 jogadores. Além disso, o jogo deveria utilizar o modelo de comunicação P2P².

Dada a situação na qual a turma se encontrava, as discussões iniciais envolveram questões relacionadas às arquiteturas do modelo P2P e qual método de envio de mensagens utilizar. Além dessas questões, houveram outras relacionadas ao comportamento do jogo relacionadas à maneira de agrupar os jogadores dispostos a jogar e o que aconteceria em caso de queda de conexão durante diferentes períodos do jogo.

Buscando solucionar as dúvidas de melhor maneira possível, foram buscadas mais informações sobre o modelo P2P e suas arquiteturas. Após ter obtido essas informações, o problema foi analisado em busca de informações que ajudassem a visualizar qual seria a arquitetura que melhor atenderia às necessidades do jogo. Devido ao problema informar que os jogadores se conectam a um servidor para encontrar outros jogadores, que as palavras a serem utilizadas devem ser decididas pelo servidor e que o ranking deve ser mantido no mesmo, foi decidido que a arquitetura híbrida atenderia melhor aos requisitos citados por integrar o P2P a uma entidade central responsável por parte dos dados.

É importante salientar que o grupo adotou uma sequência de resolução e implementação própria, sempre levando em conta as questões, fatos e ideias discutidos em sessão. Visando atender aos requisitos do problema, o grupo se dedicou primariamente a solucionar as questões relacionadas à conexão e à lógica essencial do jogo, uma vez que a *engine* do jogo encontrava-se pronta como resultado da modularização utilizada na construção do primeiro produto. Aspectos como tolerância a possíveis falhas e funcionalidades extras no jogo foram deixadas em segundo plano pelo grupo, uma vez que não são requisitos primordiais e poderiam ocupar um tempo que poderia ser gasto construindo e aperfeiçoando a interface de comunicação do jogo.

Ao verificar os métodos de envio de mensagem abordados na literatura, notou-se a existência do Unicast, Multicast e Broadcast. No jogo, além de se conectar com o seu(s) adversário(s), o player deveria se conectar com o servidor para obter os dados necessários para o início do jogo. Ou seja, dois tipos de comunicação seriam feitos, sendo que a comunicação cliente-servidor seria realizada para manifestar a intenção de jogar, e, uma vez iniciado, o jogo aconteceria utilizando comunicação P2P.

¹ Aprendizagem Baseada em Problemas

² Peer-To-Peer, ou Par-a-Par em Português.

Instrução	Descrição
16- <i>NOME-SENHA</i>	Cadastrar um novo usuário com o determinado <i>NOME</i> e <i>SENHA</i>
17- <i>NOME-SENHA</i>	Loga o respectivo jogador no servidor
18	A operação solicitada foi realizada com sucesso.
19	A operação não foi realizada ou falhou.
20- <i>MODOJOGO</i>	Aguardar por uma partida. <i>MODOJOGO</i> indica o modo de jogo.
21- <i>NOME-PONTUACAO</i>	Atualiza o ranking no servidor com as informações do vencedor.
22	Solicitar ranking

Table 1. Protocolo Cliente-Servidor

Instrução	Descrição
7-ID-VALOR	O player de id <i>ID</i> tirou <i>VALOR</i> na roleta.
8-ID-CARACTERE	O player de id <i>ID</i> jogou com o caractere <i>CARACTERE</i> .
9-X-ID1-...-IDX-NOME1-...-NOMEX-PALAVRAS-DICAS	O jogo iniciará com X jogadores, cada um deles tem a ID1.. IDX e o utilizam os nomes NOME1...NOMEX . e essas são as palavras e as dicas do jogo
10-ID	O jogador de id <i>ID</i> anunciou que o round acabou.
11-ID	O jogador de id <i>ID</i> passou a vez.

Table 2. Protocolo da comunicação P2P

A conexão com o servidor caracteriza-se como uma mensagem unicast por haver somente um destino. Pelo fato dessa característica estar presente desde o problema anterior, não houveram discussões em relação a ela. Para realizar a comunicação cliente-servidor, foi construído um protocolo baseado em texto detalhado na Tabela 1. Os itens em *itálico* são variáveis.

Uma vez conectado com o servidor, o jogo deveria ser iniciado quando haviam jogadores disponíveis em quantidade suficiente. Para iniciar a partida, os dados do jogo deveriam ser enviados do servidor para os players e os mesmos deviam ser desconectados do servidor, sendo que os players deveriam trocar informações entre si para continuar a partida. Diversas ideias surgiram, como enviar os dados do jogo para os jogadores via Unicast utilizando o protocolo de transporte TCP, enviar via Unicast utilizando o protocolo UDP e, por fim, enviar as informações do jogo via Multicast UDP. Pelo fato do jogo envolver um número de conexões considerável, o grupo escolheu utilizar o protocolo de transporte UDP, economizando recursos em comparação ao protocolo anterior. Um dos pontos negativos do protocolo escolhido é a perda de pacotes devido a latência. Sabendo que o jogo rodaria em uma rede LAN cabeada, consequentemente de baixa latência, característica que diminui a taxa de perda de pacotes, o grupo decidiu que o UDP seria o melhor protocolo para resolver o problema. Outra característica que contribuiu para a escolha do UDP foi a possibilidade de envio de mensagens via Multicast, ou seja, apenas uma mensagem seria enviada para todos os jogadores. Ao utilizar o multicast, diminuiu-se o tráfego de pacotes na rede. Em conjunto com o UDP, o Multicast fornece uma opção econômica em questão de recursos dos computadores e diminui consideravelmente a quantidade de pacotes na rede quando comparada com qualquer outra alternativa. O Multicast UDP foi utilizado para enviar as informações do servidor para os clientes e para efetuar a comunicação entre os clientes. O protocolo utilizado na comunicação P2P também foi baseado em texto e pode ser visualizado na Tabela 2.

Diversas questões surgiram em relação ao jogo. Dentre elas, destacou-se a questão

relacionada a quando o jogador deve dar a vez ao próximo quando o "Passar a Vez" for sorteado na roleta. Dentre as soluções propostas surgiram adicionar um temporizador que, ao encerrar, faria o jogador perder a vez caso não tivesse jogado ainda e, caso o jogador sinalizasse uma letra que não compusesse a palavra em questão, também passaria sua vez. Dada a prioridade da equipe de cumprir primeiramente todos os requisitos fundamentais, somente a segunda alternativa foi implementada pelo fato de exigir poucas modificações na *engine* do jogo. Além do passar a vez, destacou-se também a questão relacionada a qual dos jogadores iniciaria primeiro. As ideias que surgiram para solucionar esse impasse diziam respeito a utilizar a ordem de conexão dos jogadores para determinar quem joga primeiro, ou seja, se o jogador x foi o primeiro a entrar no jogo e solicitar um jogo em dupla, x seria quem iniciaria o jogo. Uma alternativa proposta foi que os clientes girassem a roleta sem conhecimento dos jogadores, e a ordem dos jogadores seria dada pela ordem decrescente das pontuações. Seguindo novamente a primitiva adotada pelo grupo, a ordem de conexão foi adotada como norma para definir a ordem de jogadas por ser de menor complexidade e, à vista do grupo, de mesmo impacto e mesmo fator de aleatoriedade.

Na ~~modelo~~ P2P, cada peer tem todos os dados necessários para continuar a operar. Quem cria o grupo? No caso do jogo Roletrando, cada peer possui todas as variáveis de jogo. Ao realizar uma operação, como girar a roleta, o peer atuante envia uma mensagem para o grupo de multicast informando a operação realizada e o resultado que obteve. Dessa forma, cada peer atualiza suas informações sobre o jogo. Para evitar que haja um *loop* de operações na rede, cada peer só processa a informação recebida caso o ID que conste na mensagem seja diferente do seu. Para o mesmo fim, cada peer só envia mensagens no grupo de multicast caso seja sua vez de jogar. O controle de jogadas é feito através de uma fila circular presente em cada peer. Tal fila é atualizada cada vez que uma mensagem do tipo *II-ID* é enviada ou recebida.

Ao finalizar o jogo, o vencedor do jogo deveria enviar o resultado para o servidor. Dado que cada peer sabe todas as informações do jogo, incluindo as pontuações, os peers verificam se são os vencedores e, caso não sejam, simplesmente encerram o jogo. Caso contrário, enviam a pontuação e exibem o ranking.

Por decisão do grupo, a interface foi deixada como última tarefa a ser realizada pelo fato de não ser um item essencial. Durante o processo de desenvolvimento foi implementado uma CLI³ para os testes e visualização do jogo enquanto estava em produção. Tal escolha foi feita pelo fato da CLI ser algo mais rápido de ser feito e não deixar de ser funcional. Ao finalizar todo o *background* do jogo, foi iniciada a produção da Interface Gráfica utilizando o Swing.

A interface Gráfica é composta por quatro telas:

- **Home:** Tela de configuração na qual o jogador informa um nome e a senha para se conectar ao software. Essa tela tem opção de conectar, para jogadores cadastrados e Registrar, para novos jogadores. over
- **ModoJogo:** Após o jogador se conectar no jogo, pode ser escolhido dois modos de jogo: Jogar no modo dupla e trio.
- **Game2 e Game3:** São telas semelhantes que são visualizadas a depender do modo

³Command Line Interface, ou Interface de linha de comando

jogo que o usuário informar. A mesma possibilita o jogador a jogar o Roda Roda. Estão contidos na tela dados do Jogador, como: Nome do Jogador; Valor ganho na partida atual, e dado do Jogo como palavra, dica, valor da roleta e quanto ganharam ao indicar uma letra. Além disso, a tela conta com um teclado virtual com as letras que podem ser indicadas.

É importante salientar que na interface, informações do jogo são sincronizadas em tempo real entre todos os jogadores de uma determinada partida.

4. Resultados e Discussões

Nessa seção serão abordados os principais resultados obtidos e os testes realizados no software produzido.

Antes do teste deve ser apresentado o seu resultado, ou seja, o produto.
Acho que foi semelhante ao relatório anterior...

4.1. Testes

Durante todo o desenvolvimento do software foram feitos diversos testes para garantir a integridade do produto. Os testes foram realizados em situações que simulavam o comportamento de um usuário normal através da linha de comando.

4.2. Teste da Engine do jogo

Na Engine do jogo não houve alterações referentes ao problema anterior, mas sofreu adaptações referentes à sua aplicação na lógica do jogo, anteriormente concentrada no servidor. Atualmente a Engine foi adaptada para simular o jogo no cliente, retirando a alta demanda de fluxo de dados entre servidor e cliente.

4.3. Teste de comunicação

Inicialmente, o jogo já garantia a comunicação entre cliente-servidor além de controlar o fluxo de dados entre ambos. Próximo passo era construir a interface de comunicação entre os Peers. Para testar foram construídas interface linha de comando no multicasting. Quando uma mensagem era enviada pelo multicasting, todos os clientes conectados aquele grupo exibiam a mensagem recebida na tela. Um dos principais problemas encontrados durante os testes de comunicação, foi referente a utilização do protocolo UDP, onde o mesmo não fazia controle de fluxo, mas era flexível em manter a comunicação ponto a ponto.

4.4. Teste geral do Jogo

Após os testes do jogo e de comunicação, os dois foram integrados e somados a interface gráfica. A partir desse momento foi testado o comportamento do jogo como um todo. Além de ser testada com diversos clientes e o servidor rodando na mesma máquina, também foi testado com dois ou três clientes e um servidor em máquinas diferentes. Foram identificados alguns problemas nessa etapa, após a integração da interface: Quando um jogo era finalizado o ranking não era exibido ao jogador, sendo que nos testes em linha de comando foi verificado o contrário, o Ranking estava sendo exibido sem nenhuma anormalidade. Algumas vezes quando um jogador rodava a roleta e era sorteado o “perder tudo”, o sistema zerava sua pontuação adquirida naquela rodada, o problema era na aplicação do adversário, que ao invés de zerar a pontuação do seu oponente, zerava a sua pontuação, chegando a conclusão de um erro de referência na rede.



Figure 1. Tela inicial do jogo

4.5. Resultado Final

O software “Roda a Roda”, produto final em resposta ao Problema 2 do MI - Concorrência e conectividade, simula virtualmente o jogo Roletrando. Distribuído entre 4 telas diferentes, o jogo funciona em uma rede local na qual um dos computadores atua como um servidor e os demais podem se conectar a ele para jogar. Durante o período de desenvolvimento do jogo, foi feita uma leve alteração referente ao giro da roleta, anteriormente a roleta era girada automática, agora o jogador precisa girar para poder escolher uma letra e ganhar o valor obtido. Quando um usuário cai no “Perdeu Tudo”, a interface mostra um aviso e o valor conquistado naquela rodada é perdido. As quantias das outras rodadas são mantidas como valor acumulado. Após inicializar o servidor, o jogador pode se conectar ao jogo informando o IP na qual o servidor está executando. No próximo passo o cliente deve se registrar para iniciar a jogar, caso o mesmo esteja fazendo seu primeiro login. Para fazer login, o jogador deve informar o nome e a senha feita no em seu respectivo cadastro, como demonstrado na figura 1. Depos de fazer o login, o jogador pode escolher dois modos de jogo: dupla ou trio, como demonstrado na figura 2. O jogo só começa quando o número de jogadores for correspondente aos modos, mostrando a cada jogador a tela demonstrada na figura 3. após a finalização do jogo o ranking não está sendo exibido por problemas até o momento desconhecidos, já que o mesmo é visualizado pelo teste em linha de comando. O software desenvolvido conta com uma interface simples e amigável ao usuário, buscando tornar o jogo o mais fácil de entender e jogar.

5. Conclusão

Através das análises, discussões e estudos foi-se desenvolvido o jogo que, através de testes manuais, teve resultado parcialmente satisfatório, pois, o programa é capaz de executar os modos de jogo requisitados, sendo esses, dupla e trio, através de uma interface interativa e com clareza, compartilhando as informações necessárias, com exceção de uma boa gerencia de pontuações em virtude de problemas na opção Perdeu Tudo da roleta, para o bom funcionamento das partidas entre os jogadores. Outro requisito falho é a exibição do ranking de maiores pontuações, o qual não foi possível ser implementado.

Analizando a produção, foi-se pensado na adição de músicas de fundo para tornar o jogo mais cativante, melhorando ainda mais o cenário virtual. Os estudos dos con-

Parágrafo longo. Deve escrever dando pausa...

Primeiro deve dizer o que fez, depois descrever os problemas. Deve enaltecer o que foi feito!!

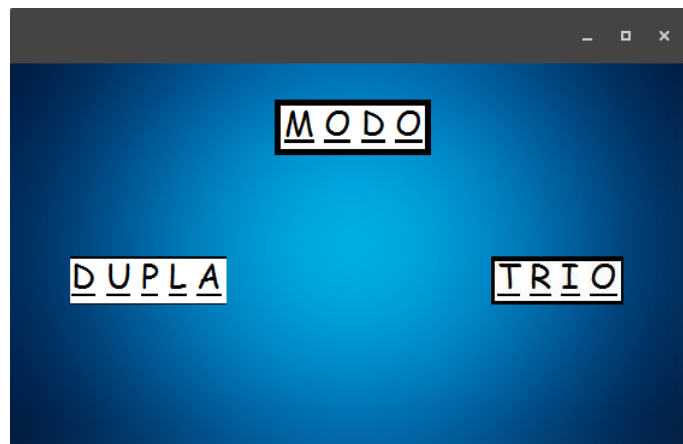


Figure 2. Tela da escolha da modalidade de jogo.



Figure 3. Tela principal do jogo.

ceitos usados para produzir esse software são de suma importância pois são essenciais ao desenvolvimento e entendimento de sistemas aplicados a redes de computadores

References

Forouzan, B. A. (2009). *Comunicação de dados e redes de computadores*. AMGH Editora.

Rocha, J., Domingues, M., Callado, A., Souto, E., Silvestre, G., Kamienski, C., and Sadok, D. (2004). Peer-to-peer: Computação colaborativa na internet. *Minicurso, Simpósio Brasileiro de Redes de Computadores*.

Rosenzweig, P., Kadansky, M., and Hanna, S. (1998). The java reliable multicast service: A reliable multicast library. *Sun Microsystems Laboratories SML Technical Report Series*.

Tanenbaum, A. S. (2003). *Redes de computadores*. Pearson Educación.