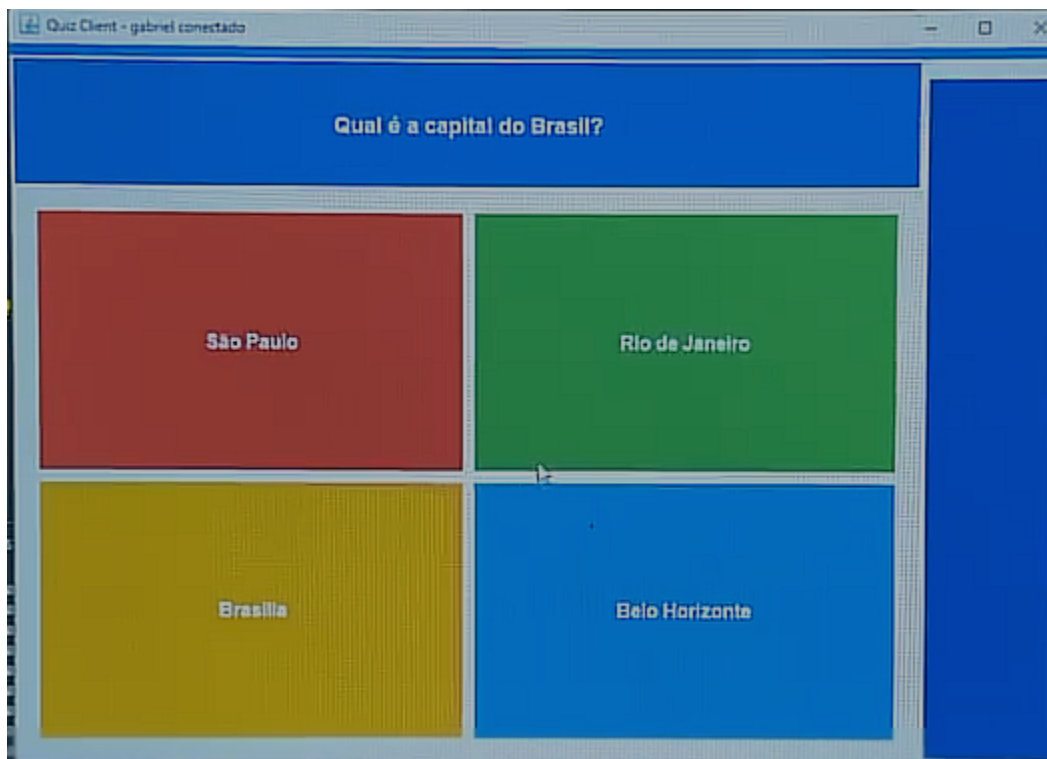


Relatório do Trabalho Prático - Jogo Quiz Competitivo (Estilo Kahoot)

Redes de Computadores I

Ana Fernanda Souza
Arthur de Sá Braz
Gabriel Praes Bernardes Nunes
Guilherme Otávio de Oliveira
Júlia Pinheiro Roque
Vitória Símil Araújo

Este trabalho apresenta o desenvolvimento de uma aplicação de rede baseada em um jogo de perguntas e respostas competitivo, inspirado no Kahoot!. A aplicação desenvolvida simula essa dinâmica ao desafiar os participantes com perguntas de conhecimentos gerais, promovendo uma competição entre os jogadores. A cada resposta enviada, o servidor avalia se está correta e atualiza um placar em tempo real, permitindo que os participantes acompanhem seu desempenho ao longo do jogo. O sistema de pontuação considera tanto a correção quanto o tempo de resposta, estimulando agilidade e precisão. Além do aspecto interativo, o projeto foi construído para demonstrar conceitos práticos de redes de computadores, como a comunicação via protocolos TCP e UDP, multithreading no servidor para atender múltiplos clientes simultaneamente, e a configuração de rede com roteadores, endereçamento IP, NAT e roteamento entre sub-redes distintas.



O intuito do trabalho é colocar em prática os conceitos aprendidos em sala de aula. Entre seus objetivos, destacam-se:

- Desenvolver aplicação cliente-servidor utilizando Java
 - Implementar comunicação TCP para dados críticos
 - Implementar comunicação UDP para atualizações em tempo real
 - Utilização de multithreading para interface responsiva
 - Configuração de rede com roteadores reais e faixas de IP específicas
 - Demonstrar funcionamento de port forwarding e roteamento
-

Metodologia

A metodologia adotada para o desenvolvimento do trabalho foi dividida em quatro etapas principais: análise dos requisitos da aplicação, definição da arquitetura de rede, implementação em Java com técnicas de multithreading e protocolos de comunicação, e testes em uma topologia real utilizando roteadores físicos.

As funcionalidades da aplicação ajudaram a guiar decisões técnicas como qual protocolo utilizar (TCP/UDP) e quais seriam as responsabilidades por threads e incluem:

- Interface Gráfica Amigável para Jogadores
- Conectividade via rede
- Envio de Perguntas com opções de resposta
- Recebimento de respostas em tempo real
- Atualização do placar e do tempo restante

A arquitetura da aplicação foi projetada para funcionar como **cliente-servidor**, onde o **servidor** gerencia o fluxo do jogo, envia perguntas e controla o placar e os **clientes** se conectam ao servidor, visualizam as perguntas e enviam suas respostas.

Além disso a aplicação foi construída utilizando a linguagem Java com:

- **Swing** para a interface gráfica (GUI), com telas de conexão e de jogo
 - **Socket e DatagramSocket** para comunicação TCP e UDP, respectivamente
 - **Multithreading**, com uma thread dedicada para escutar mensagens TCP, outra para UDP, e a thread principal para interface gráfica.
-

Interface de Redes

A comunicação em rede da aplicação foi implementada utilizando os protocolos TCP e UDP, combinando confiabilidade e desempenho.

O protocolo **TCP** foi escolhido para o envio e recebimento de informações críticas, como as perguntas, as respostas dos jogadores, a sincronização inicial e o placar, garantindo a entrega ordenada e sem perdas. Já o protocolo **UDP** foi utilizado para transmissões que exigem baixa latência como notificações de ações dos jogadores, aceitando eventuais perdas de pacotes sem comprometer a lógica do jogo, como detalhado na tabela abaixo:

Protocolo	Utilização	Justificativa
TCP	Comunicação Confiável Utilização	Garante que dados essenciais como perguntas e respostas sejam entregues corretamente e na ordem adequada, mantendo a integridade do jogo.
UDP	Atualizações em Tempo Real Utilização	Permite atualizações com latência mínima. Como essas informações são temporárias, a eventual perda de pacotes não compromete a lógica principal.

Cada cliente estabelece uma conexão TCP com o servidor e, paralelamente, escuta mensagens UDP em uma porta dedicada. No lado do servidor, uma thread específica é responsável por enviar essas atualizações via UDP para todos os clientes conectados. Essa divisão estratégica entre os protocolos permite à aplicação manter consistência nas informações essenciais, ao mesmo tempo em que oferece uma experiência fluida e em tempo real durante as partidas.

Multithreading

Servidor:

- Thread Principal (GUI): Interface Swing e interação com administrador
- Thread TCP Listener: Aceita novas conexões de clientes
- Thread UDP Sender: Envia atualizações periódicas

- Threads por Cliente: Uma thread para cada cliente conectado

Cliente:

- Thread Principal (GUI): Interface Swing e interação do usuário
- Thread TCP Listener: Recebe mensagens do servidor
- Thread UDP Listener: Recebe atualizações em tempo real

Para sincronização e Thread Safety as estruturas incluíram `ConcurrentHashMap` para armazenar clientes conectados, `SwingUtilities.invokeLater()` para acesso aos dados compartilhados e métodos sincronizados para acessos compartilhados.

Configuração da Rede

Equipamentos	Conexões
PC1 → Cliente	LAN R1
PC2 → Servidor	LAN R3
R1 (classe B)	PC1
R3 (classe A)	PC2

Além disso, a WAN do R1 foi conectada à LAN do R3.



DIR-615

Assistente

Rede Local

Configuração da Internet

IPv6

Configuração

Wireless

Avançado

Maintenance

Status

Ajuda

Configuração Da Interface LAN

Esta página é usada para configurar a interface LAN do seu Roteador Wireless. Aqui você pode alterar a configuração do endereço IP, máscara de sub-rede, etc.

Esta página pode ser utilizada para configurar o modo de DHCP: Nenhum ou servidor DHCP.

(1) Habilitar o servidor DHCP se você estiver utilizando este dispositivo como um servidor DHCP. Esta página lista os grupos de endereços IP disponíveis para os hosts na sua rede local. O dispositivo distribui números do grupo aos hosts na sua rede conforme eles solicitam acesso à Internet.

Se você selecionar "Nenhum", então o roteador não fará nada quando os hosts solicitam um endereço IP.

(2) Esta página lista o endereço IP fixo/MAC na sua LAN. O dispositivo distribui o número configurado aos hosts na sua rede conforme eles solicitam acesso à Internet.

Configuração Da Interface LAN

Endereço IP: 10.0.0.1

Máscara de Sub-rede: 255.0.0.0

Aplicar mudanças

Configurações De Servidores DHCP

Modo DHCP: DHCP Server

Faixa do grupo de Ips: 10.0.0.2 - 10.255.255.254

Tempo máx. de aluguel: 120 minutos

Nome de domínio: domain.name

Servidor DNS 1: 10.0.0.1

Servidor DNS 2: (Optional)

Aplicar mudanças

Desfazer

Dicas úteis...

O endereço IP do seu roteador é o mesmo endereço IP que você usará para acessar a interface de gerenciamento web do seu roteador.

Se você já tem um servidor DHCP na sua rede ou se estiver usando endereços IP estáticos em todos os dispositivos da sua rede, selecione o Modo DHCP Nenhum para desativar este recurso.

Se você possui dispositivos na sua rede que sempre devem ter endereços IP fixos, adicione um DHCP estático para cada dispositivo deste tipo.

Mais...

TP-LINK

Archer C3200

Quick Setup

Basic

Advanced

Logout

Factory

Status

Network

Internet

LAN

IPv6

DHCP Server

Dynamic DNS

Advanced Routing

IPSec VPN

Wireless

Guest Network

NAT Forwarding

USB Settings

Parental Controls

Bandwidth Control

WAN Interface

Internet Connection Type: Static IP

IP Address: 10.0.0.2

Subnet Mask: 255.0.0.0

Default Gateway: 10.0.0.1 (Optional)

Primary DNS: 8.8.8.8 (Optional)

Secondary DNS: 0.0.0.0 (Optional)

Advanced

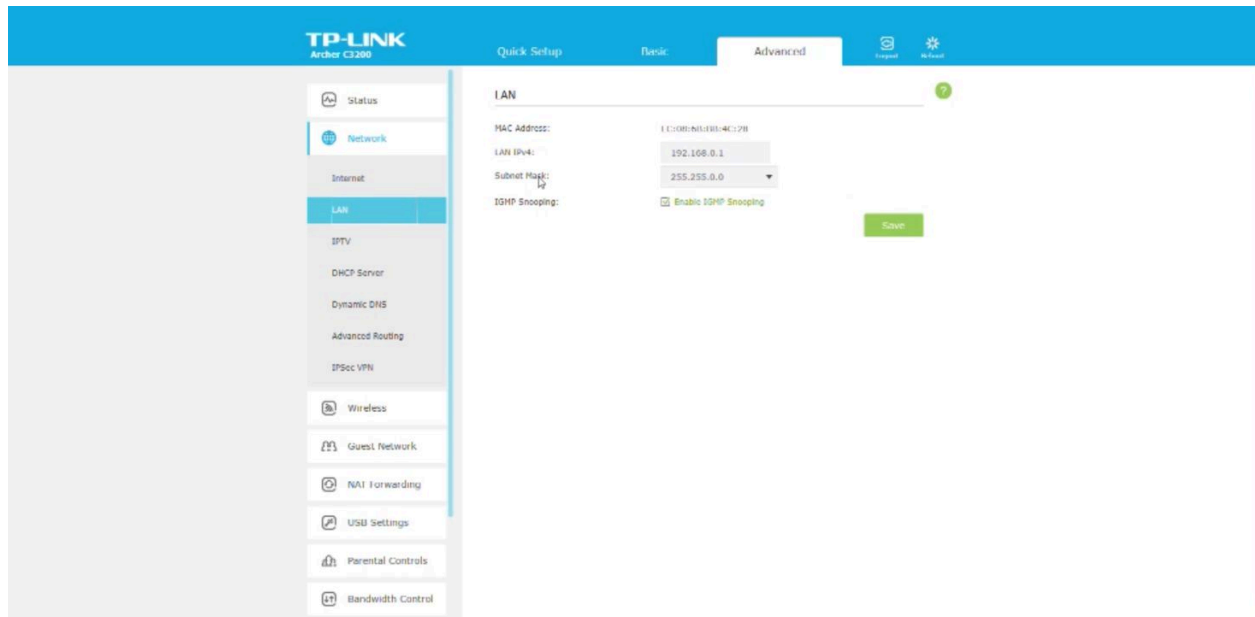
MAC Clone


Use Default MAC Address

Use Current Computer MAC Address

Use Custom MAC Address


Save




PC1 (Cliente)
IP: 192.168.0.10
Gateway: 192.168.0.1


PC2 (Servidor)
IP: 10.0.0.10
Gateway: 10.0.0.1


R3 (classe A)
IP da LAN: 10.0.0.1
Máscara: 255.0.0.0


R1 (classe B)
IP da LAN: 192.168.0.1
Máscara: 255.255.0.0

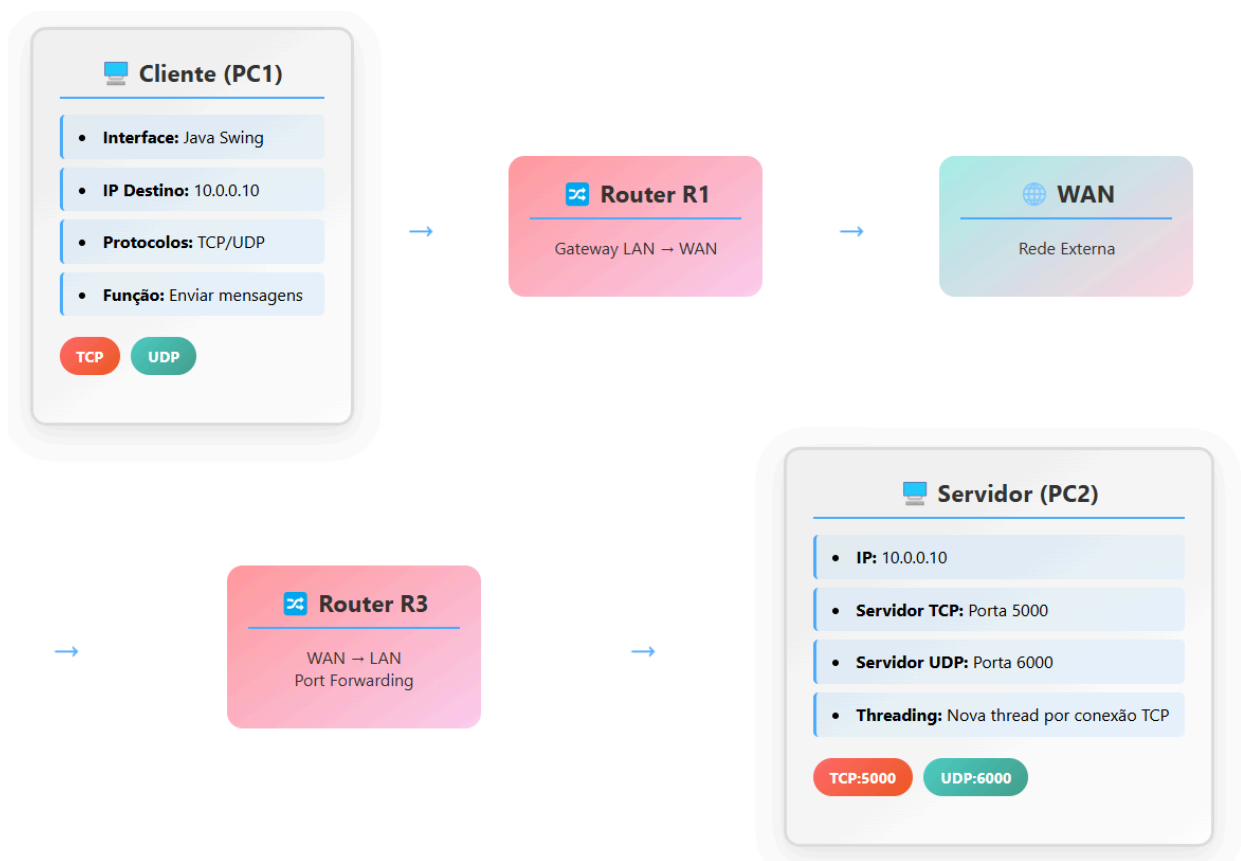
WAN do R1:
IP: 10.0.0.2
Gateway: 10.0.0.1 (IP do R3)

O roteador R3 é quem conecta as redes, e por isso foi configurado para:

- Redirecionar portas (Port Forwarding):
- Porta TCP 5000 → 10.0.0.10
- Porta UDP 6000 → 10.0.0.10

Isso permite que mensagens enviadas do cliente (PC1) cheguem ao servidor (PC2), mesmo que eles estejam em redes diferentes.

Fluxo da Aplicação



Servidor (PC2):

- Inicia dois serviços:
- Um servidor TCP escutando na porta 5000
- Um servidor UDP escutando na porta 6000
- Cada nova conexão TCP é atendida por uma nova thread

Cliente (PC1):

- Interface Java Swing
 - Envia dados para o IP 10.0.0.10 (PC2), nas portas correta
-

Conclusão

Em conclusão, a implementação do jogo permitiu consolidar na prática diversos conceitos fundamentais de redes de computadores, como comunicação via TCP/UDP, programação concorrente com multithreading e configuração de redes físicas.

A aplicação demonstrou-se funcional, com desempenho satisfatório tanto em ambiente simulado quanto em uma rede real e entre os principais desafios enfrentados, destaca-se a configuração manual dos roteadores e o estabelecimento da comunicação entre dispositivos em sub-redes distintas, exigindo atenção especial à definição de rotas, NAT e regras de port forwarding. Apesar dessas dificuldades, a equipe conseguiu superar os obstáculos, resultando em uma aplicação estável, interativa e alinhada com os objetivos propostos pela disciplina.