

Projeto de Rede - Warbugs

- Equipe:
- Eder Ferreira
- Eduardo Fantini
- Paulo Henrique Abreu
- Tarcísio Rezende

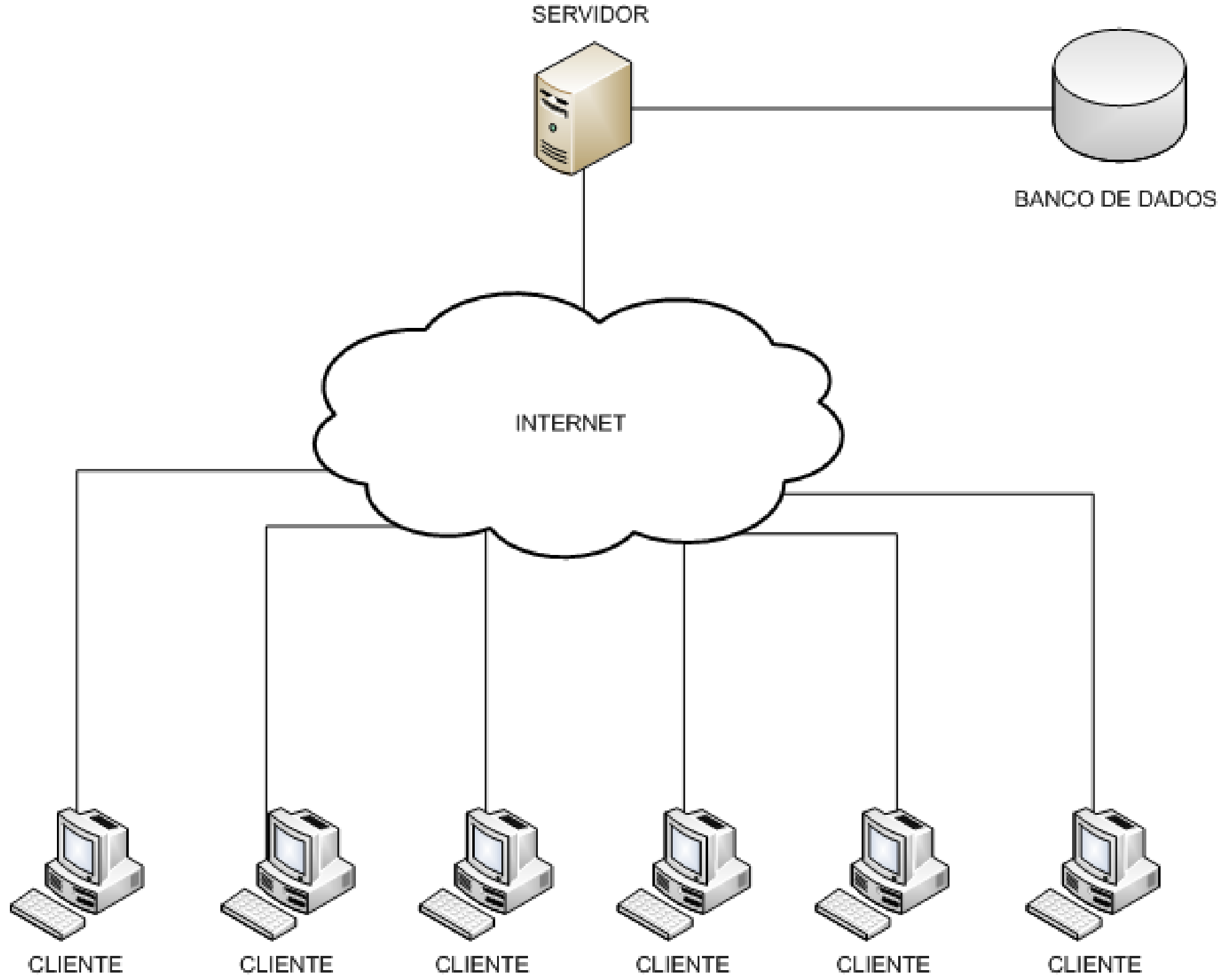


Roteiro

- Introdução
- Cliente
- Servidor
- Protocolo
- Dicionário de mensagens
- BugMessage
- Aspectos do jogo
- Problemas e Soluções
- Trabalhos Futuros

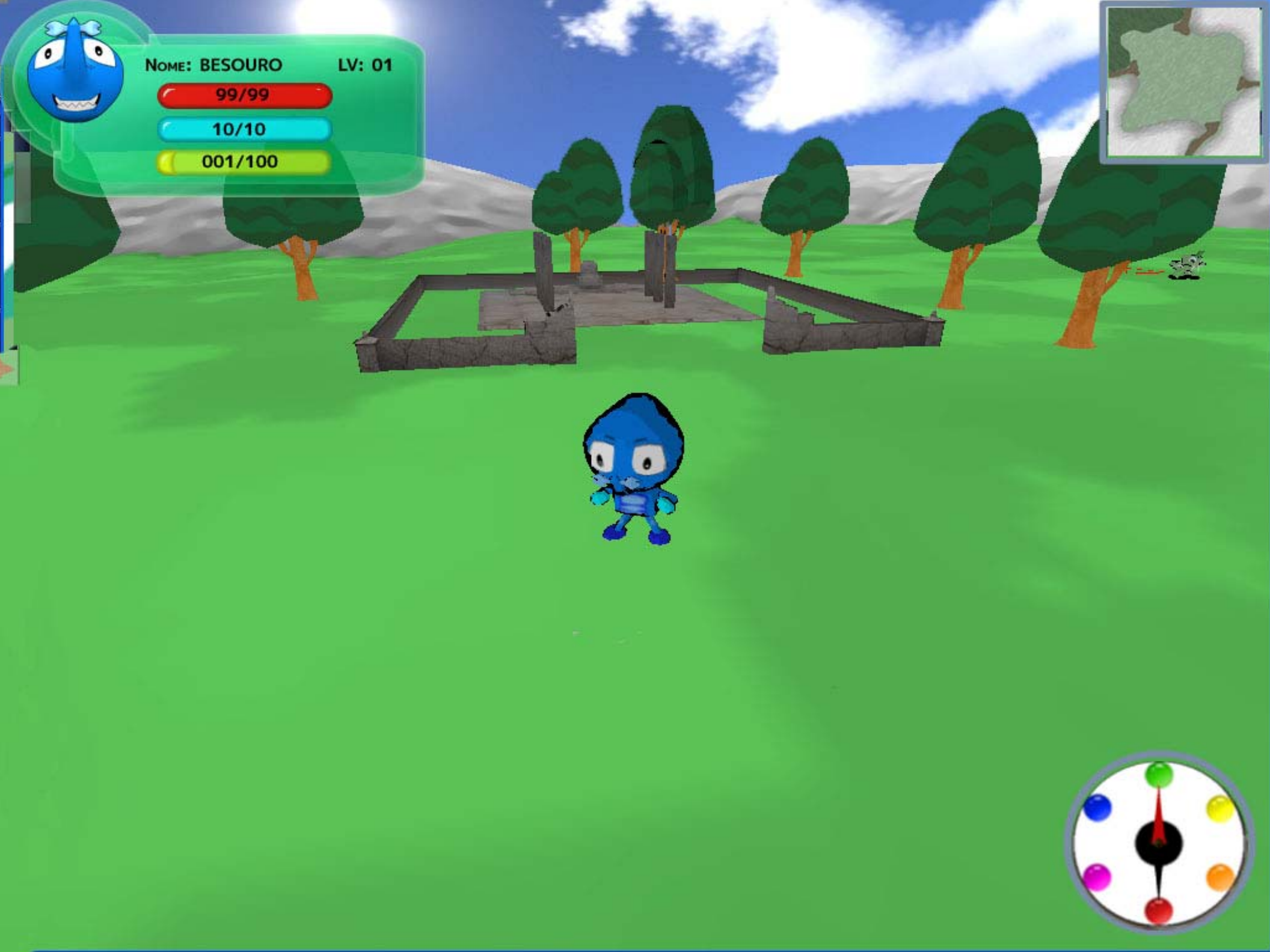
Introdução

- O projeto de redes foi desenvolvido para o trabalho de conclusão de curso Warbugs.
- Um RPGMO (Role-Playing Game Multiplayer Online).
- Topologia cliente/servidor.



Aplicação Cliente

- Loop por frame com métodos sequenciais:
 - Recebe os pacotes do servidor
 - Atualiza os dados dos objetos na cena
 - Renderiza
 - Processa os comandos do jogador
 - Envia pacotes para o servidor
 - Taxa de 30 FPS



NOME: BESOURO

LV: 01

99/99

10/10

001/100



Aplicação Servidor

- Funcionamento por frame:
 - Recebe os pacotes dos clientes
 - Atualiza os objetos do jogo
 - Armazena as mensagens a enviar em um buffer
 - Envia as mensagens para os clientes específicos
- Mundo persistente
- Mercado econômico persistente

Configurações Iniciais

FPS

Valor Atual: 0

Novo Valor:

30

Aplicar

Tempo de Verificação de Ping

Valor Atual (Segundos): 0

Novo Valor:

010

Aplicar

Valor de Tolerância de Ping

Valor Atual (Segundos): 0

Novo Valor:

005

Aplicar

Configurações de Saves

Tempo de Backup Total

Valor Atual (Horas): 0

Novo Valor:

00

Aplicar

Tempo de Backup de Mercado

Valor Atual (Minutos): 0

Novo Valor:

00

Aplicar

Protocolo TCP

- Garantia de entrega
 - As mensagens trafegam em meio instável
- Controle de fluxo de dados
- Controle de congestionamento
 - Aumento da qualidade do serviço
- BugSocket
 - Possui timeout e não fica em espera no listening

BugMessage

- Cria uma sequência de bytes
- Dados de entrada e saída serializados em bytes:
 - Ex.: inteiro = message.readInt();
 - Ex.: message.writeInt(inteiro);
- Torna o tamanho de pacote dinâmico

Dicionário de Mensagens

- Cada tipo de mensagem um ID e seus parâmetros (opcionais). Ex:

CREATE_PERSONAGEM

int idUsuario, int idRaca, char * nome

- As mensagens não são otimizadas e não possuem criptografia.

Aspectos do Jogo

- Sincronização e acurácia altos devido à taxa de atualização por frames
- Regras do jogo dificultam o game exploit
- Escalabilidade projetada: 50 jogadores
- Baixa tolerância a falhas

Problemas e soluções

- Multi-Thread
 - Assíncronas
 - Necessita de gerenciamento para sessões críticas (semáforo)
- Solução
 - Game loop com métodos sequenciais

Problemas e soluções

- Problemas com o disconnect
 - Mensagem de desconexão do cliente não chega ao servidor
- Solução
 - Sleep de 1 milisegundo antes de fechar o socket.

Problemas e soluções

- Banco de Dados
 - Biblioteca do mySQL não aceita procedures
- Solução
 - Implementar as procedures manualmente

Trabalhos Futuros

- Criptografia
 - Criptografar mensagens críticas (login, trocas, comércio, etc)
- Balanceamento de Carga
 - Implementar balanceamento de carga de servidores e sincronização dos mesmos

Trabalhos Futuros

- Otimização de pacotes:
 - Unificar mensagem de erro e adicionar um parâmetro para diferenciação
 - Otimização de variáveis no formato de bits
- Otimização do servidor:
 - Para aumentar sua escalabilidade