JOGO ADEDONHA / STOP ENTRE COMPUTADORES

Jogo de adedonha entre computadores que se comunicam pela rede.

Cursos: Engenharia e Ciência da Computação.

Disciplina: Sistemas Distribuídos e Tolerância de Falhas.

Professor: Fábio Contarini Carneiro.

Alunos:

João Pedro Lima Garcia (1200201054),

Isabelle Vitória Dias Ramos de Oliveira (1210201774),

Lucas Vieira da Cunha Oliveira (1210202215),

Anne Besant da Costa Dutra (1210200301),

Danilo Cavalcante Mensor (1210201671).

Rio de Janeiro

Outubro, 2024

Descrição do Jogo

O jogo stop/adedonha funciona criando uma lista com diversas categorias, cidade, objeto, cor e comida, por exemplo, esses categorias são previamente definidos pelos participantes da brincadeira, após listar, cada jogador coloca um número e os números são somados, após a soma dos números, é verificado que letra no alfabeto é equivalente ao número, por exemplo se a soma dos números for 4, a letra a ser escolhida é D.

Após a listagem dos categorias e definição de qual letra foi escolhida, todos os integrantes devem simultaneamente preencher a lista sendo a resposta com a inicial da letra selecionada.

Quando um dos integrantes preenche todos os categorias, ele grita “STOP”, fazendo isso todos devem mostrar a sua lista preenchida independente se estão completas ou não, ao fazer isso é validado se todos os categorias da lista foram preenchidos corretamente considerando que foram iniciados com a letra, que estejam gramaticalmente corretos e que façam sentido com o item.

A contagem dos pontos é feita de acordo com as palavras diferentes, se ninguém que está participando digitou uma palavra igual, essa palavra vale 10 pontos por exemplo, mas caso uma ou mais pessoas tenham escolhido a mesma palavra no mesmo item, cada um vai receber 5 pontos, caso todos cheguem à conclusão de que a palavra não faz sentido com o item, essa pessoa recebe 0 pontos.

Arquitetura Utilizada na Implementação

A arquitetura utilizada foi a Cliente-Servidor, amplamente empregada em sistemas distribuídos. Nessa arquitetura, o servidor central é responsável por gerenciar todas as interações e processar as solicitações dos clientes, que, por sua vez, se conectam ao servidor para realizar operações específicas, como enviar respostas ou receber atualizações do jogo.

O servidor (ou host) é o núcleo do sistema e é implementado em Python, no servidor que é configurado qual endereço deve procurar e em qual porta vai ser conectado. No host também ocorre a espera pela conexão do cliente, envia a letra

sorteada e a lista de categorias para cada usuário, recebe as respostas dos clientes e processa as respostas.

O usuário (ou cliente) também rodam códigos em Python, eles são os jogadores, cada cliente se conecta ao servidor, envia suas respostas e aguarda feedback do servidor para validar se estão corretas ou não.

Essa arquitetura distribui o processamento de dados de forma centralizada no servidor, o que garante que as regras do jogo sejam aplicadas de maneira consistente, sem divergências entre os jogadores.

Como foi Estruturada a Solução do Sistema Distribuído

A solução foi implementada para rodar de forma distribuída, permitindo que múltiplos jogadores se conectem de diferentes computadores para participar do jogo. Essa distribuição foi possível utilizando sockets, que permitem a comunicação entre máquinas conectadas a uma rede, seja ela local ou através de uma VPN (neste caso, criada com o Hamachi).

O servidor é configurado para escutar em uma porta da rede e esperar por conexões por parte do cliente, cada vez que um cliente se conecta o servidor cria uma nova thread para tratar a comunicação com aquele cliente, permitindo múltiplas conexões.
O cliente se conecta ao servidor através de um endereço de IP e porta, após a conexão o usuário recebe a letra sorteada e a lista de categorias que devem ser preenchidas, após serem preenchidas as palavras os dados são enviados de volta ao servidor e recebe a mensagem de confirmação.

Funcionamento do Protocolo com Explicação das Mensagens Trocadas

Na nossa aplicação estamos usando o protocolo TCP/IP, esse protocolo é confiável, orientado a conexão, garantindo a entrega dos dados na ordem correta, evitando duplicação, o que é crucial porque as mensagens precisam chegar na ordem certa e sem alterações em sua estrutura.

As mensagens trocadas entre os usuários (clientes) consistem na solicitação do preenchimento das categorias de animal, país e objeto, todos os campos separados por vírgula. Após o preenchimento e envio da mensagem, o usuário que preencheu vai receber a mensagem “Respostas registradas! Aguarde os outros jogadores.” e vai ser desconectado, após todos os usuários responderem o servidor devolve a mensagem que todos os usuários preencheram, no formato de lista.

Explicação de Como Compilar e Executar o Programa

Todo o projeto foi desenvolvido em Python, logo, tanto os jogadores (clientes) quanto o servidor devem ter o código instalado na máquina, assim como um compilador de código para facilitar na edição e execução do código, mesmo que seja rodado em um terminal, outro programa que precisa ser instalado é uma ferramenta de rede privada virtual (VPN), que no caso do nosso projeto foi utilizado o programa Hamachi, para que todos estejam conectados na mesma rede.

No computador que vai servir de servidor (host), deve ser executado o código em Python *servidor.py*, neste código também deve ser alterado o HOST e o PORT de acordo com o IP configurado na VPN, após a execução desse código basta aguardar a conexão dos clientes.

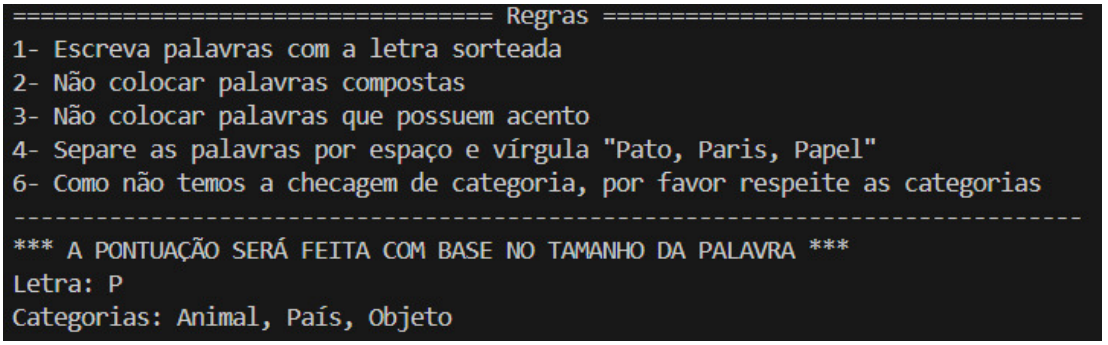
O computador usado pelos jogadores (clientes), vai executar o código em Python *cliente.py*, também alterando o HOST e o PORT para que se conectem na rede criada, caso a execução tenha sucesso o cliente vai receber a letra e as categorias para preencher.

A execução funciona da seguinte forma:

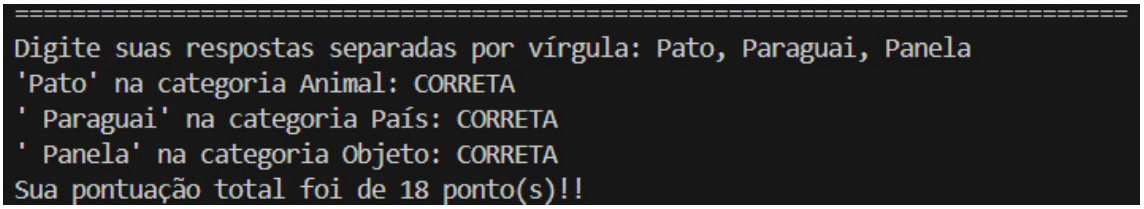
1. O servidor é iniciado em um dos computadores e fica aguardando a conexão dos jogadores (clientes).
2. Cada cliente se conecta executando seu código, após executar recebe a letra e as categorias, preenche as respostas e envia de volta ao servidor, ao enviar as respostas as conexões são encerradas.
3. O servidor recebe as respostas de todos os clientes e retorna na tela do host a resposta que cada jogador deu.

Prints de Tela Mostrando o Funcionamento do Programa

Tela do jogador (cliente) ao executar o código e ter sucesso na conexão.



Tela do jogador (cliente) ao enviar as respostas, retornando o feedback se as respostas estão corretas ou não.



Programas Utilizados

Para conexão de rede entre as máquinas foi utilizado o programa Hamachi, esse programa cria redes, permitindo a criação e participação em redes privadas virtuais, onde cada computador conectado atua como se estivesse na mesma rede local, mesmo que estejam fisicamente separados.

Para desenvolvimento do código em Python foi usado o compilador VSCode, no mesmo compilador foi utilizado o console para fazer a comunicação entre as máquinas.

Também utilizamos o GitHub como plataforma de controle de versão e colaboração, o repositório foi fundamental para facilitar o compartilhamento de código entre os membros da equipe, garantindo que todos tivessem acesso à versão mais atualizada do projeto, além disso, o GitHub permitiu a criação de um histórico das alterações feitas no código, promovendo um fluxo de trabalho organizado e colaborativo, com cada membro contribuindo de forma eficiente para diferentes partes do sistema. O link para o Git é o: *https://github.com/isabellediasr/Jogo-Adedonha-Sistemas-Distribuidos*.