



UNIVERSIDADE VEIGA DE ALMEIDA

Ciência da computação

Daniel Alves da Cunha – 1210100434
Julia Victoria Carvalhosa de Amorim – 1210202404
Mariana Muniz da Rocha – 1210201822
Mateus Restier de Sousa Noronha – 1210202324
Pedro Gomes Gonçalves – 1210202324

Sistemas Distribuídos e Tolerância de Falhas

A1

RIO DE JANEIRO – RJ
Outubro/2024

Sumário

Introdução.....	3
Descrição do jogo	3
Arquitetura utilizada na implementação	3
Solução de Sistema Distribuído	3
Funcionamento do protocolo e mensagens trocadas	4
Compilação e execução do programa.....	4
Conclusão.....	5
Bibliografia	6

Introdução

O presente trabalho tem como objetivo a implementação de um jogo utilizando conceitos de sistemas distribuídos. Os sistemas distribuídos são amplamente utilizados em diversas áreas da tecnologia moderna, onde a comunicação entre diferentes dispositivos é essencial para o funcionamento das aplicações. Neste trabalho, será implementado um jogo de adivinhação de palavras chamado “Palavra enforcada, no qual dois jogadores, em diferentes computadores, interagem via rede para jogar, utilizando um modelo cliente-servidor.

O jogo desenvolvido exemplifica os princípios de construção de programas distribuídos e a troca de mensagens entre diferentes participantes.

Descrição do jogo

Nome do Jogo: Palavra Enforcada

Objetivo: O objetivo do jogo é que o jogador cliente tente adivinhar a palavra secreta escolhida pelo jogador servidor, antes que o número máximo de erros seja atingido (enforcamento).

Regras:

- O jogo envolve dois jogadores.
- Um jogador escolhe a palavra secreta e atua como servidor.
- O outro jogador tenta adivinhar as letras da palavra, e se conectar ao servidor via socket.
- O jogador tem um número máximo de 5 tentativas erradas.
- A cada tentativa, o jogador pode chutar uma letra, que será verificada no servidor.
- O jogo termina quando o jogador adivinha a palavra ou atinge o número máximo de erros (enforcamento).

Arquitetura utilizada na implementação

O jogo foi implementado utilizando uma arquitetura cliente-servidor baseada em sockets. O servidor é responsável por armazenar a palavra secreta e processar os chutes do jogador. O servidor se conecta ao cliente via TCP/IP, e recebe a palavra secreta via socket.

A comunicação entre as entidades utilizando a biblioteca padrão socket. O servidor escuta em uma porta predefinida e espera a conexão do cliente, enquanto o cliente envia a palavra secreta.

Solução de Sistema Distribuído

A solução foi implementada como um sistema distribuído utilizando comunicação via rede. O servidor aguarda uma conexão TCP do cliente e recebe

uma palavra secreta. Após essa fase, o cliente se conecta novamente e envia os chutes ao servidor, que compara a letra chutada com a palavra secreta e retorna se a letra está correta ou não.

O jogo foi implementado para tolerar erros como a digitação de letras repetidas, oferecendo feedback adequado ao jogador. Além disso, o servidor conta o número de erros para determinar quando o jogador perdeu o jogo.

O protocolo de comunicação utilizado é o TCP, garantindo a confiabilidade no envio e recebimento de dados entre o servidor e o cliente.

Funcionamento do protocolo e mensagens trocadas

As mensagens trocadas entre o cliente e o servidor seguem o seguinte fluxo:

1. O servidor inicia e aguarda a conexão de um cliente.
2. O cliente se conecta ao servidor e envia a palavra secreta.
3. Após a palavra ser recebida, o servidor encerra a conexão com o cliente.
4. O jogo prossegue localmente no servidor, sem mais troca de mensagens entre cliente e servidor.
5. O jogador realiza os chutes diretamente no servidor, que processa as tentativas e determina o progresso do jogo (acertos, erros, vitória ou derrota).

Compilação e execução do programa

Pré-requisitos:

- Python 3.x
- Biblioteca padrão socket

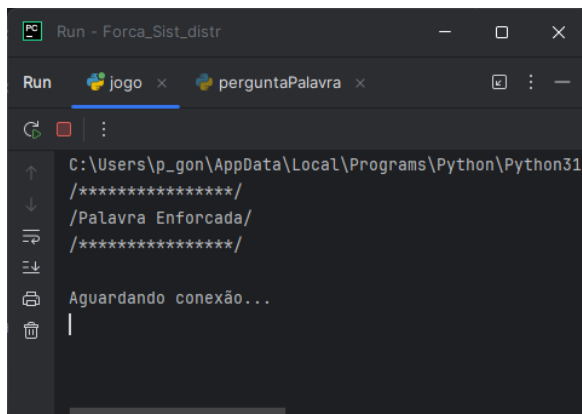
Passos para compilar e executar (considerando rodar cliente e servidor na mesma máquina):

1. **Servidor:**
 - Execute o arquivo `jogo.py`.
 - O servidor ficará esperando pela conexão do cliente e pela palavra secreta.
2. **Cliente:**
 - Execute o arquivo `perguntaPalavra.py`.
 - O cliente enviará a palavra secreta para o servidor e, depois disso, iniciará o jogo.

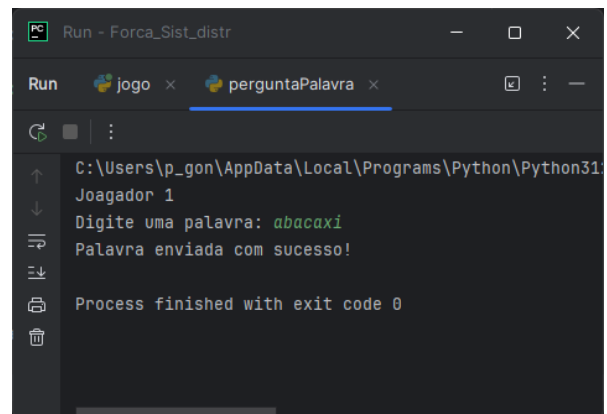
Conclusão

O desenvolvimento deste trabalho permitiu a compreensão prática dos conceitos de sistemas distribuídos, como a comunicação via rede entre diferentes dispositivos. A implementação de um jogo simples como o "Palavra Enforcada" foi uma oportunidade para explorar técnicas de programação distribuída e tolerância a falhas em um contexto de diversão e aprendizado.

Capturas de tela

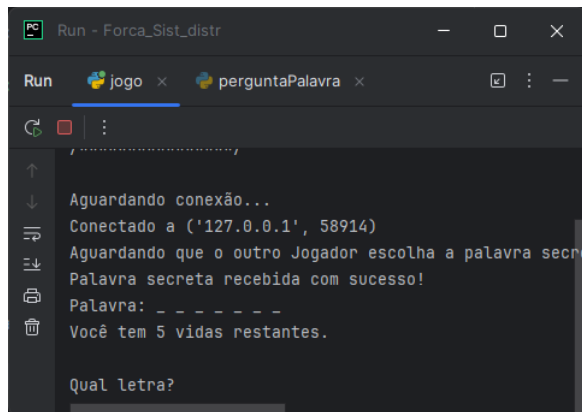


```
Run - Forca_Sist_distr
Run  jogo x perguntaPalavra x
C:\Users\p_gon\AppData\Local\Programs\Python\Python312\
/*****/
/Palavra Enforcada/
/*****/
Aguardando conexão...
|
```



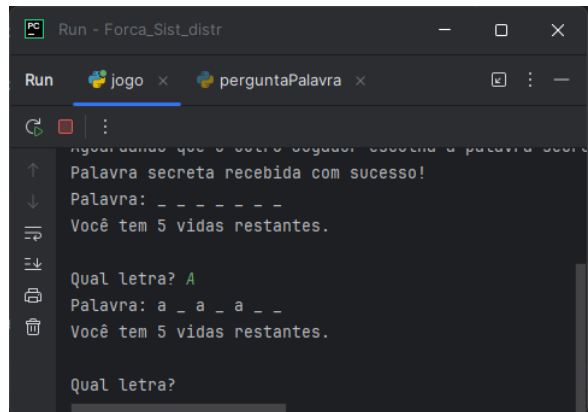
```
Run - Forca_Sist_distr
Run  jogo x perguntaPalavra x
C:\Users\p_gon\AppData\Local\Programs\Python\Python312\
Joagador 1
Digite uma palavra: abacaxi
Palavra enviada com sucesso!

Process finished with exit code 0
```



```
Run - Forca_Sist_distr
Run  jogo x perguntaPalavra x
Aguardando conexão...
Conectado a ('127.0.0.1', 58914)
Aguardando que o outro Jogador escolha a palavra secreta
Palavra secreta recebida com sucesso!
Palavra: _ _ _ _ _
Você tem 5 vidas restantes.

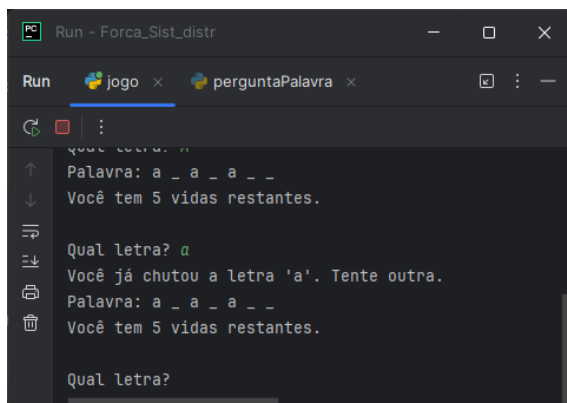
Qual letra?
```



```
Run - Forca_Sist_distr
Run  jogo x perguntaPalavra x
Palavra secreta recebida com sucesso!
Palavra: _ _ _ _ _
Você tem 5 vidas restantes.

Qual letra? A
Palavra: a _ a _ a _ _
Você tem 5 vidas restantes.

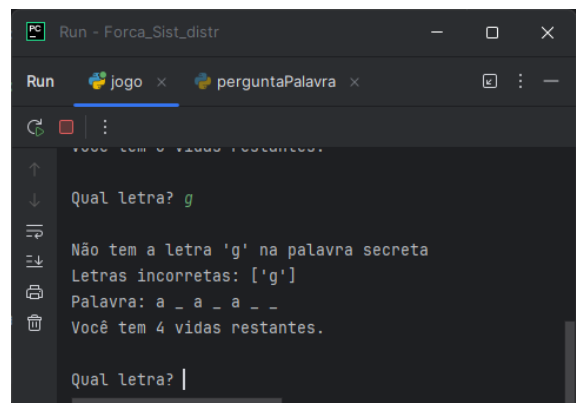
Qual letra?
```



```
Run - Forca_Sist_distr
Run  jogo x perguntaPalavra x
Palavra: a _ a _ a _ _
Você tem 5 vidas restantes.

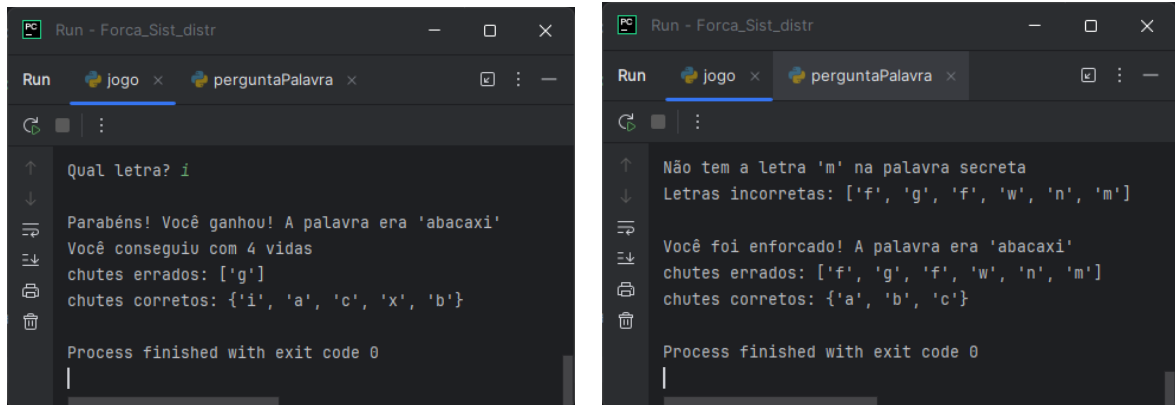
Qual letra? a
Você já chutou a letra 'a'. Tente outra.
Palavra: a _ a _ a _ _
Você tem 5 vidas restantes.

Qual letra?
```



```
Run - Forca_Sist_distr
Run  jogo x perguntaPalavra x
Qual letra? g
Não tem a letra 'g' na palavra secreta
Letras incorretas: ['g']
Palavra: a _ a _ a _ _
Você tem 4 vidas restantes.

Qual letra? |
```



Bibliografia

PYTHON. **socket** — Low-level networking interface — Python 3.8.1 documentation. Disponível em: <<https://docs.python.org/3/library/socket.html>>.

VAN STEEN, M. **DISTRIBUTED SYSTEMS Second Edition**. [s.l.: s.n.]. Disponível em: <https://vowi.fsinf.at/images/b/bc/TU_Wien-Verteilte_Systeme_VO_%28G%C3%B6schka%29_-_Tannenbaum-distributed_systems_principles_and_paradigms_2nd_edition.pdf>.