TÍTULO DO RELATÓRIO

EMPRESA

Autor1, Autor2

VERSAO

TÍTULO DO RELATÓRIO

DEPARTAMENTO

EMPRESA

Autor1, Autor2
(nmec1) autor1@ua.pt, (nmec2) autor2@ua.pt

DATA

Resumo

Este projeto foi realizado no âmbito da cadeira LABI (acr) do 1º ano do MIECT. Consiste em criar um servidor em que os clientes se conectam para jogar um jogo de adivinha o número secreto. Para além disso, também tivemos que fazer este mesmo relatório em que nós explicamos o projeto: objetivo, motivação, a metodologia utilizada, resultados e conclusões. Na metodologia, será relatado em pormenor o nosso código feito para criar o servidor, tanto código do cliente como do servidor, o modo de funcionamento, testagem e comandos git feitos para tal. Nos resultados, será mostrado o fruto de todo o nosso código que é o servidor a funcionar. Por fim, nas conclusões, retira-se o que se alcançou com este projeto, o que aprendemos, o quão útil este projeto é para compreendermos esta matéria da cadeira de LABI e o quão interessante foi realizá-lo.

Agradecimentos

Eventuais agradecimentos. Comentar bloco caso não existam agradecimentos a fazer.

Índice

Capítulo 1

Introdução

O objetivo do jogo é que o cliente adivinhe um número secreto entre 0 e 100 criado aleatoriamente pelo servidor, dentro de um determinado número de tentativas dadas também pelo servidor. O cliente tem a possibilidade de realizar quatro operações diferentes. A primeira serve para iniciar o jogo (START), a segunda serve para tentar adivinhar o número secreto (GUESS), a terceira serve para desistir do jogo (STOP) e a última serve para sair (QUIT). O servidor deve responder adequadamente aquando da operação pretendida. Se houver algum erro, como um cliente inexistente, o servidor deve dar uma resposta adequada ao cliente. No final do jogo, o servidor guarda num ficheiro .csv os dados do cliente que jogou o jogo. Para efeitos de segurança, o cliente possui ainda a possibilidade de escolher se quer que os seus dados sejam encriptados ou não.

Este documento está dividido em quatro capítulos. Depois desta introdução, no Capítulo 2 é apresentada a metodologia seguida, no Capítulo 3 são apresentados os resultados obtidos, sendo estes discutidos no Capítulo 4. Finalmente, no Capítulo 5 são apresentadas as conclusões do trabalho.

Capítulo 2

Metodologia

2.1 Cliente

Nesta secção será apresentada a metodologica no ficheiro client.py

2.1.1 Função main

Devido ao facto do cliente ser invocado com o comando, esses python3 client.py client_id porto [máquina]"argumentos têm de ser validados. Para começar, devem ser colocados 3 ou 4 argumentos (máquina é opcional). Caso não tenha nenhuma destas quantidades, é enviada uma mensagem de erro. Caso o tamanho seja 3, a máquina é a local(127.0.0.1), caso seja 4 a máquina é o último argumento.

O client_id não possui qualquer tipo de restrição, logo a única verificação feita é se ele existe (len(argv[1])). Para a porta, existem 2 condições: a porta inserida é constituída apenas por números e esse número encontra-se entre 0 e 65535. Para isso, percorre-se todos os caractéres da string e, caso não seja um dígito, é enviada uma mensagem de erro.

Por fim, a máquina tem de ser verificada, os seus números entre cada "."estão compreendidos no intervalo]0, 255]. Os números são colocados num array através do método ".split('.')"e, caso algum não satisfaça a condição, é enviada mensagem de erro.

Após esta verificação, é criado o socket com a porta e máquina indicados na invocação, tenta-se estabelecer conexão e chama-se a função run_client, que é onde se vai passar o jogo. Quando terminar, é fechado o socket e o cliente termina.

2.1.2 Função run_client

A função run_client é invocada na main. O cliente é introduzido ao jogo e é-lhe questionado se pretende encriptação de dados (S/N). Enquanto a respotas for diferente dessas duas letras, é enviada uma mensagem de erro. Após

a inserção da opção, é criado um dicionário start com o id e a cifra do cliente. Se foi inserido "S", é criada uma chave que é inserida no dicionário start na chave cipher. É usada a função sendrecv_dict do common_comm como recomendado para enviar start e receber a resposta do servidor(recvstart). É verificado se houve algum erro no início do jogo através de validate_response, se houve termina, senão a variável maxAttempts fica com o valor correspondente em recvstart(desencriptada caso necessário).

2.1.3 Função validate_response

A função validate_response procura por uma chave "error"no dicionário response, que corresponde à resposta de um servidor ao que foi enviado pelo cliente. Visto que, quando existe um erro, seja qual for a operação, esta chave é enviada, é feita essa procura e, se existir, é enviado um valor booleano True, caso contrário é enviado False

2.1.4 Função quit_operation

Nesta função, é criado um dicionário quit com uma única chave com o nome da operação "QUIT". O dicionário recvquit é criado para receber a resposta do servidor ao enviar quit através de sendrecv_dict. Se a função validate_response verificar que existe um erro, é enviado o return da chave do erro. Se não houver erro, o cliente é informado que desistiu depois de x tentativas, sendo x "attempts"

2.2 Servidor

2.3 Exemplos

Capítulo 3

Resultados

Descreve os resultados obtidos.

Capítulo 4

Análise

Analisa os resultados.

Capítulo 5

Conclusões

Com este trabalho, conseguimos solidificar o nosso conhecimento de servidores em python, sockets, interações entre cliente e servidor e criação de algoritmos para tal, bem como encriptação e desencriptação de dados para uma partilha de informação mais segura. Houve também uma aproximação à linguagem Python e a toda à sua sintaxe e características. Sendo Python uma linguagem com bastante procura no mercado de trabalho, a criação do servidor veio ajudar a compreender a sua autenticidade e pas suas diferenças em relação a outras linguagens com que já estamos habituados (Ex: Java) Apesar das adversidades, acreditamos que o trabalho foi conseguido com sucesso, conseguimos criar um servidor com o jogo referido e com todas as características necessárias para tal, utilizando os recursos que nos foram dados e auxiliando a sua compreensão com este relatório.

Contribuições dos autores

Resumir aqui o que cada autor fez no trabalho. Usar abreviaturas para identificar os autores, por exemplo AS para António Silva. No fim indicar a percentagem de contribuição de cada autor.

Acrónimos