# Spyke: App de Mensagens

## Equipe:

Eduardo Buzzi e Lucas Teixeira.

## Descrição do projeto:

Spyke é um aplicativo de mensagens disponível apenas para sistemas desktop. Nele, é possível criar sua conta própria e conversar com qualquer usuário da plataforma. Inspirado na versão oficial “Skype”, o Spyke é um programa que está sendo desenvolvido para fins acadêmicos com base na linguagem Java utilizando Sockets para conexão cliente-servidor. É possível ver todos que estão online na plataforma, e mandar mensagens para qualquer um. Envios de arquivos estão sendo planejados para realizar dentro do app. As mensagens compartilhadas entre cliente-servidor são formatadas em JSON, que facilita bastante a montagem dos objetos em diversas extremidades da conexão. Inicialmente não possui recurso para imagens de perfil.

## Requisitos Funcionais:

|  |  |
| --- | --- |
| RF01 | O aplicativo deve permitir cadastro de novos usuários. |
| RF02 | O aplicativo deve permitir usuários realizarem login. |
| RF05 | O aplicativo deve permitir a visualização de todos os contatos. |
| RF06 | O aplicativo deve permitir ver quem está online. |
| RF07 | O aplicativo deve permitir adicionar amigos. |
| RF08 | O aplicativo deve permitir enviar mensagem de texto para os contatos. |

## Requisitos não Funcionais:

|  |  |
| --- | --- |
| RNF01 | As senhas dos usuários devem ser criptografadas. |
| RNF02 | O app deve ser disponível para usuários Desktop. |

## Especificação das mensagens trocadas entre cliente e servidor:

A manipulação dos dados é realizada através de mensagens que são formatadas em JSON, para facilitar a montagem dos objetos. Como os dados estão sendo definidos e em constante atualização, abaixo há um exemplo de como é realizado o login:

|  |  |
| --- | --- |
| **Conteúdo da mensagem** | {"action":1,"inputs":["eduardo","54d56s4w8d4d5s"]} |
| **Descrição** | A variável ‘action’ contém o número de acordo com a operação, 1 sigifica LOGIN. A variável inputs contém os campos que estão sendo enviados para o servidor, nesse caso, usuário e senha. |
| **Retorno** | {"action":1,"status":"success"}  É retornado um JSON com a status definida “success” se o login foi validado, caso contrário o status é definido como “fail”. |

REGISTER:

|  |  |
| --- | --- |
| **Conteúdo da mensagem** | {"action":0,"inputs":[“Eduardo”,"eduardo","54d56s4w8d4d5s"]} |
| **Descrição** | A variável ‘action’ contém o número de acordo com a operação, 0 sigifica REGISTER. A variável inputs contém os campos que estão sendo enviados para o servidor, nesse caso, nome, usuário e senha. |
| **Retorno** | {"action":0,"status":"success”}  É retornado um JSON com a status definida “success” se o login foi validado, caso contrário o status é definido como “fail”. |

ADD CONTACT:

|  |  |
| --- | --- |
| **Conteúdo da mensagem** | {"action":2,"inputs”:[ “eduardo”]}} |
| **Descrição** | A variável ‘action’ contém o número de acordo com a operação, 0 sigifica ADD CONTACT. A variável inputs contém os campos que estão sendo enviados para o servidor, nesse caso, usuario. |
| **Retorno** | {"action":2,"status":"success”}  É retornado um JSON com a status definida “success” se o login foi validado, caso contrário o content é definido como “fail”. |

REMOVE CONTACT:

|  |  |
| --- | --- |
| **Conteúdo da mensagem** | {"action":3,"inputs”:[ “eduardo”]}} |
| **Descrição** | A variável ‘action’ contém o número de acordo com a operação, 0 sigifica remove CONTACT. A variável inputs contém os campos que estão sendo enviados para o servidor, nesse caso, usuario. |
| **Retorno** | {"action":3,"status":"success”}  É retornado um JSON com a status definida “success” se o login foi validado, caso contrário o content é definido como “fail”. |

GET CONTACTS:

|  |  |
| --- | --- |
| **Conteúdo da mensagem** | {"action":9,"inputs”:[ 1]} |
| **Descrição** | A variável ‘action’ contém o número de acordo com a operação, 0 sigifica get CONTACT. A variável inputs contém os campos que estão sendo enviados para o servidor, nesse caso, id usuario. |
| **Retorno** | {"action":9,"status":"success”, “content”: “{[“eduardo”, “lucas”]}”}  É retornado um JSON com os usernames de todos os amigos com a status definida “success” se o login foi validado, caso contrário o content é definido como “fail”. |

GET USER:

|  |  |
| --- | --- |
| **Conteúdo da mensagem** | {"action":10,"inputs”:[“eduardo”]} |
| **Descrição** | A variável ‘action’ contém o número de acordo com a operação, 0 sigifica get user. A variável inputs contém os campos que estão sendo enviados para o servidor, nesse caso, usuario. |
| **Retorno** | É retornado um JSON com com as informações do usuário solicitadas, se não for encontrado, a o status será definido “fail” e o contente é vazio. |

TEXT MESSAGE:

|  |  |
| --- | --- |
| **Conteúdo da mensagem** | {"action":4,”usernameSender”: “eduardo”,”content”: “Conteudo da mensagem”} |
| **Descrição** | A variável ‘action’ contém o número de acordo com a operação, 4 sigifica TEXT MESSAGE. USERNAMESENDER terá o nome de usuário de quem esta enviando. |
| **Retorno** | É retornado um JSON com todas as informações enviadas para o cliente do outro lado da conexão. |