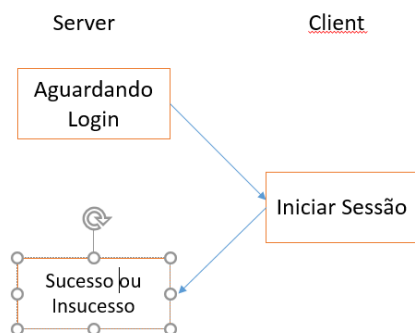


• Introdução

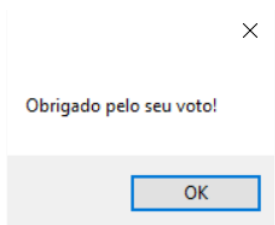
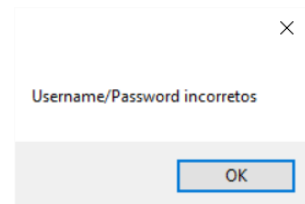
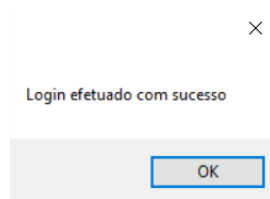
A presente aplicação tem como objetivo realizar um sistema de votações. Para a elaboração desta votação, é necessário que ocorra uma ligação entre um servidor e um cliente, através de um acesso na qual os processos enviam e recebem mensagens, isto é, através de um Socket. Este Socket assíncrono faz com que não seja possível haver dois votadores ao mesmo tempo, nem dois servidores conectados ao mesmo tempo, visto que estamos a utilizar só uma porta

Outro conteúdo necessário a abordar é o processo de comunicação no protocolo TCP que tem a capacidade de colocar um servidor num porto definido e que respeita as conexões desse porto. O cliente deve saber antecipadamente o IP ou nome do servidor e o porto em específico onde o servidor está situado à espera das conexões.

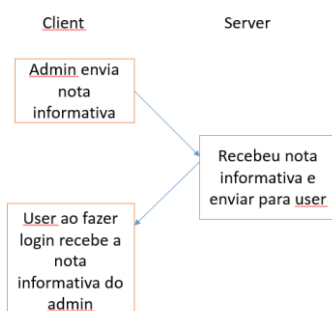
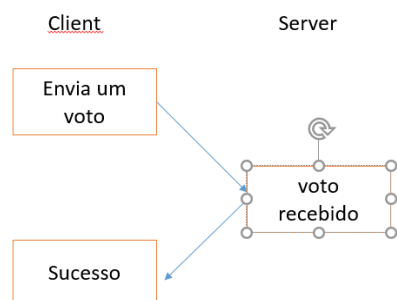
• Formato das mensagens da aplicação



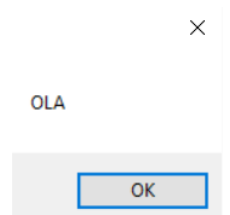
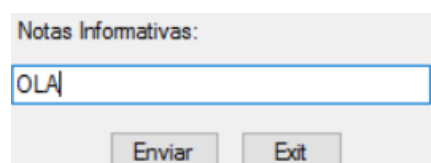
Quando o servidor é iniciado aguarda enquanto o cliente inicia sessão. Ao iniciar sessão como admin ou user é mostrada uma mensagem a indicar que a sessão foi efetuada com sucesso ou não.



Quando é iniciada a sessão como user é fornecida a votação de momento e após escolher a opção pretendida é indicado que foi enviado um voto. Não pode haver voto nulo, por defeito a primeira vem selecionada.



Quando é iniciada a sessão como admin é possível enviar notas informativas ao user, isto é, o admin ao enviar a nota informativa passa-a pelo servidor onde fica guardada até o user voltar a iniciar sessão.



• Tratamento de falhas na comunicação

Existem falhas de comunicação tanto do lado do cliente como do lado do servidor, no entanto a única em comum faz com que não seja possível iniciar o Socket por algum motivo desconhecido.

```
        catch (Exception e) //Exceções
        {
            Console.WriteLine(e.ToString());
        }
    }
```

As restantes falhas de comunicação estão no lado do cliente:

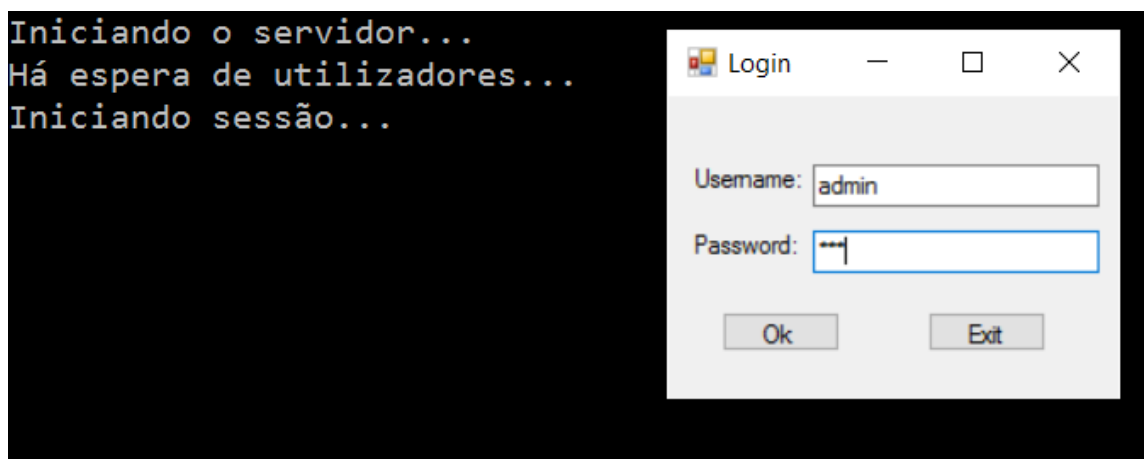
```
/*a)*/catch (ArgumentNullException ane) //Exceções
{
    Console.WriteLine("ArgumentNullException : {0}", ane.ToString());
}
/*b)*/catch (SocketException se)
{
    Console.WriteLine("SocketException : {0}", se.ToString());
}
/*c)*/catch (Exception e)
{
    Console.WriteLine("Unexpected exception : {0}", e.ToString());
}
```

- “/*a)*/” - Exceção de quando uma referência que não pode ser nula e é passada como parâmetro.
- “/*b)*/” - Exceção de quando o Socket não é conectado com sucesso.
- “/*c)*/” - Exceção inesperada por qualquer outro motivo.

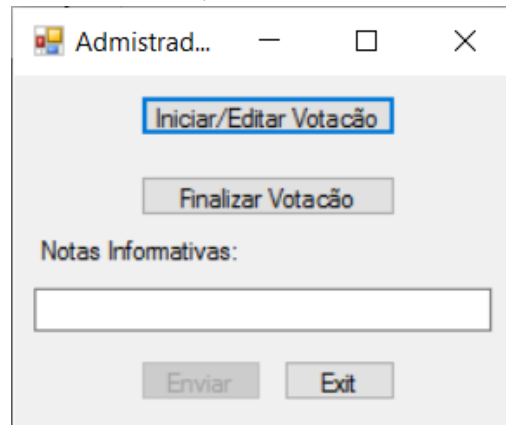
• Manual de utilização/ testes efetuados à aplicação

Teste número 1 (como administrador da votação):

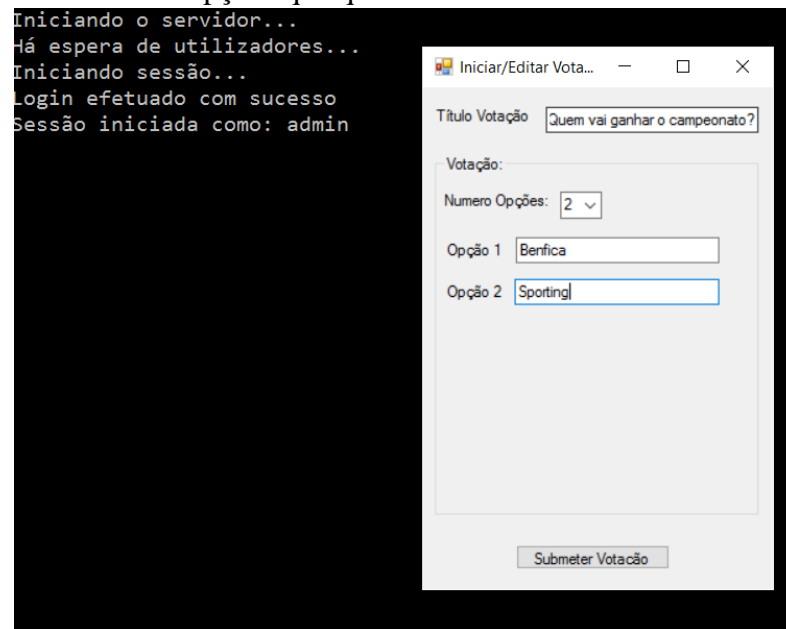
- a) Primeiramente é necessário iniciar votação como admin, sendo a password abc;



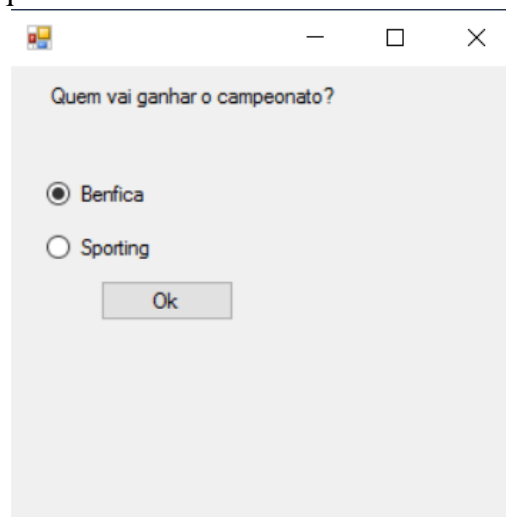
- b) Após o login realizado com sucesso temos 3 opções: editar a votação, finalizar a votação e enviar notas informativas;



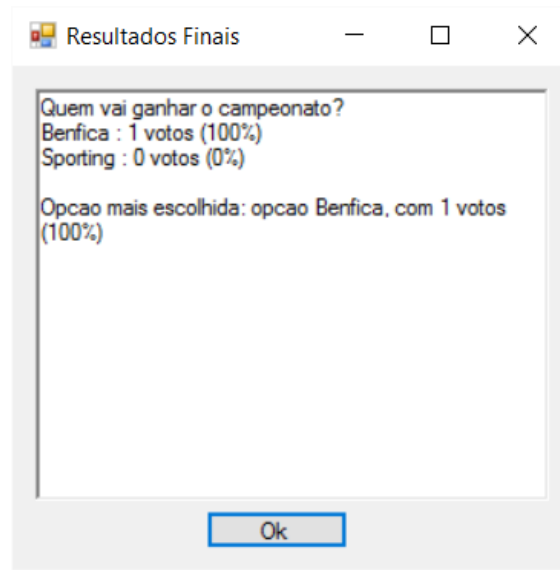
- c) Se optarmos por iniciar/editar a votação podemos decidir qual a pergunta, as opções e o número das opções que queremos.



- d) Após submeter a votação se iniciarmos sessão como user podemos verificar que são estas as opções que lá constam.

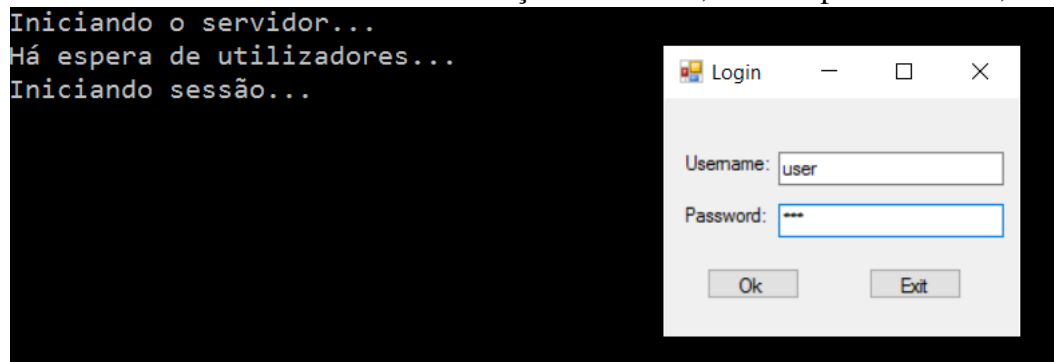


- e) Se optarmos pela finalização da votação existente, não vai ser possível receber mais votos e são mostrados o total de votos, respetivas percentagens e o item vencedor.

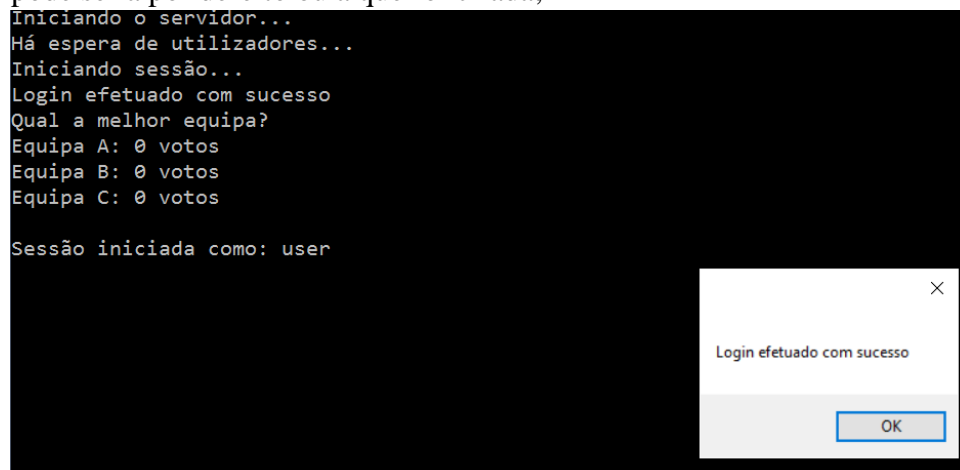


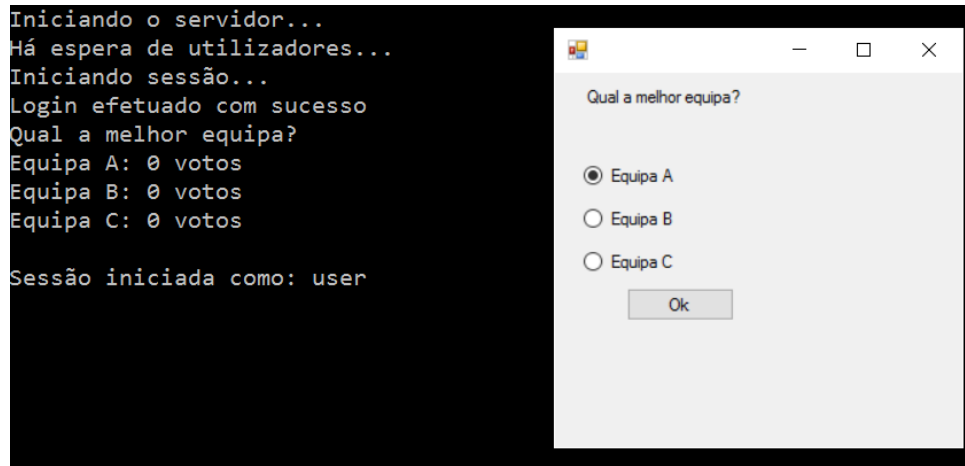
Teste nº2 (como votador):

- Primeiramente é necessário iniciar votação como user, sendo a password 123;



- Quando o login é efetuado corretamente o votante vota na opção pretendida que pode ser a por defeito ou a que foi criada;





- Ao carregar OK verificamos que o voto foi contabilizado.

```
Sessão iniciada como: user  
Voto para: Equipa A  
Estado votacao: Qual a melhor equipa?  
Equipa A: 1 votos  
Equipa B: 0 votos  
Equipa C: 0 votos  
  
Há espera de utilizadores...
```