Redes de Computadores

Projeto
2018 / 2019

Leiloeira Fanecas

Universidade Lusófona de Humanidades e Tecnologias
Licenciatura em Engenharia Informática
Docentes: Miguel Tavares, José Faísca
Discentes: Francisco Silva 21705328, Rodrigo Cassanheira 21703091

Índice

Registo de licitadores

Para registar licitadores foi criado uma thread no regulador que chama um método que executa um ciclo infinito. Neste ciclo, o regulador pede ao utilizador o username do licitador e verifica se já existe algum licitador com esse username. Caso exista um licitador com esse username o regulador pede novamente ao utilizador que insira um username, caso contrário, o regulador pede ao utilizador que insira uma password. Após o utilizador inserir a password, o regulador transforma a password numa hash através do algoritmo SHA-256. De seguida o regulador pede ao utilizador que insira o plafond do licitador.

Finalmente, após estes dados serem introduzidos e validados, o regulador instância um objeto da classe Licitador, insere-o numa lista de objetos desta classe e reescreve o ficheiro dos licitadores com todos os objetos desta lista (esta parte é explicada mais à frente).

Guardar e ler de ficheiro as credenciais dos licitadores

Para os licitadores foi implementado uma classe Licitador que tem as credenciais como atributos e implementa a interface Serializable. Esta classe é bastante importante uma vez que todos o licitadores existentes estão inseridos numa lista de objetos desta classe.

Para ler e guardar as credenciais dos licitadores foi implementado a classe Serializador que tem apenas duas funções, uma que permite serializar para um ficheiro e outra que permite deserializar de um ficheiro. Estas funções são genéricas uma vez que a função serializar recebe um objeto da classe Object e a função deserializar também devolve um objeto da classe Object.

O regulador, ao ser iniciado, a primeira coisa que faz é ler os ficheiros, isto inclui o ficheiro de licitadores. Esta é a única situação em que o ficheiro de licitadores é lido. No regulador foi criado um método para ler o ficheiro de licitadores que chama a função deserializar da classe Serializador para deserializar o ficheiro de licitadores. Uma vez desserializado é necessário dar cast do objeto da classe Object para a lista de objetos da classe Licitador.

Para guardar as credenciais, foi criado um método no regulador que chama a função serialize da classe Serializador para serializar em ficheiro a lista de licitadores. Este método para serializar é chamado sempre que se regista um licitador, quando uma licitação é validada e sempre que um leilão termina.

Guardar as passwords através de uma hash

Como foi dito anteriormente no passo de Registo de licitadores “Após o utilizador inserir a password, o regulador transforma a password numa hash através do algoritmo SHA-256” o objecto da classe Licitador fica com a hash no atributo password e desta forma quando a lista de objetos da classe Licitador é serializada em ficheiro, a password já é hash.

Autenticação de utilizadores

Com efeito de manter o projeto organizado foi feito uma herança da classe Pedido. A classe Autenticação é uma das classe que herda da classe Pedido, e através desta classe sabemos o username e a password do licitador que fez o pedido.

No regulador existe um método que recebe um socket. Este método lê a mensagem do socket e define que tipo de pedido é neste caso seria o de autenticação, assim sendo seria redirecionado para um método que recebe o objeto Autenticação e o socket.

Se o utilizador que fez o pedido Autenticação estiver na lista posteriormente desserializada e os dados que introduziu correspondem aos mesmos que se encontram nessa lista o regulador confirma a autenticação caso contrário o regulador nega a autenticação.

Criação de novos leilões a pedido de licitadores

A classe PedidoCriarLeilao é outra classe que herda da classe Pedido, através desta classe sabemos o username do autor do leilão, o objeto a leiloar, o valor inicial do leilão e a data de fecho.

Usando o mesmo método do regulador referido acima que trata de pedidos, o pedido é redirecionado neste caso para o método que recebe o objecto PedidoCriarLeilao, através de um ciclo é encontrado o autor do leilão da lista de objetos Licitador e o leilão é criado e adicionado à lista de objetos Leilão.

Notificar o autor da criação do leilão com sucesso

Após o passo acima, é chamado um método que recebe a mensagem que será enviada pelo regulador e o address do autor do leilão. Este método irá criar um pacote que irá ser enviado para o Autor da criação do leilão.

Notificar todos os licitadores da criação de novos leilões

No seguimento do passo descrito acima, é feito um ciclo à lista licitadores onde cada licitador é informado através do método que recebe a mensagem da criação de novos leilões, e o address do licitador. Este método irá criar um pacote que irá ser enviado para cada licitador exceto o autor do leilão.

Pedido da lista de leilões a decorrer

Através do método que trata de pedidos já referido em autenticação de utilizadores, é redirecionado para outro método que irá fazer um ciclo na lista com objetos Licitador para determinar quem fez o pedido de seguida irá fazer outro ciclo na lista com objetos Leilão do regulador e irá concatenar na mensagem todos os leilões existentes ao momento. Após este passo irá chamar o método que recebe a mensagem e o address do requisitante do pedido e cria o pacote que irá ser enviado para este licitador.

Realização de licitações

A classe Licitação é outra classe que herda da classe Pedido, através desta classe sabemos o username do licitador, a quantia licitada e o id do leilão licitado.

Usando o mesmo método referido acima que trata de pedidos, é redirecionado para outro método que irá fazer um ciclo na lista de objectos Leilão para encontrar o leilão licitado, e onde se irá desenvolver o resto das validações da licitação que irão ser referidas abaixo.

Regulador verifica os critérios de validação das licitações

Dentro do ciclo anteriormente referido existirá outro ciclo que determinará quem fez a licitação, dentro deste ciclo existem condições para determinar se a quantia licitada é maior que a maior licitação atual. Se sim, então a maior quantia passa a ser a licitada e esta quantia licitada é colocada na lista de objetos Licitação da classe Leilão.

Notificar ao licitador proposta aceite

Se as condições acima se verificarem, então, irá ser feito um ciclo para determinar o licitador com a maior Licitação. Após ser determinado é chamado o método que recebe a mensagem, neste caso, a licitação foi aceite e ainda recebe o address do licitador. Este método irá criar um pacote que é enviado para o licitador.

Notificar ao licitador proposta não aceite

Se as condições da validação das licitações não se verificarem, então, irá ser feito um ciclo para determinar o licitador desta proposta. Após ser determinado é chamado o mesmo método acima descrito, mas, neste caso, recebe a mensagem que a licitação não foi aceite.

Consistência do plafond dos licitadores

Sempre que se valida a licitação de um leilão é feito um ciclo para determinar o licitador da antiga maior licitação. Após este ser determinado é chamado o método que recebe a antiga maior licitação e acrescenta-a ao plafond do licitador correspondente.

Ao licitador da atual maior licitação é lhe retirada a licitação feita ao plafond.

Notificações de fim de leilão

No regulador existe um método que está constantemente a percorrer a lista de Objectos Leilão. Dentro deste ciclo cada leilão chama o método para ver se a data atual é posterior à data de terminar o leilão se sim então o leilão termina (o boolean *terminado* atributo da classe Leilão passa a true) .

De seguida é criada uma thread que vai fechar o leilão enviando as notificações a todos os participantes deste chamando um método do regulador que recebe um leilão. Este método inicialmente vê se o leilão teve licitações através de um método na classe leilão que vê se o hashset de participantes está vazia (a classe Leilão tem um hashset do nome dos Licitadores) de seguida num ciclo que percorre todos os licitadores caso o utilizador tenha participado no leilão e este tenha feito a maior licitação é lhe enviada a mensagem de felicitação de ter ganho o leilão através do método já descrito anteriormente método este que recebe a mensagem e o adress do licitador. Caso o utilizador tenha só participado no leilão mas não o ganhou é enviado através do mesmo método descrito acima mas com uma mensagem de participação. Se os leilões tiverem o hashset de participantes vazio então é enviado para o autor do leilão a notificação de lamentação através do mesmo método que envia notificações.

Notificação de uma nova licitação

Usando o método para a realização de licitações na classe regulador e após as validações da licitação já referidas acima, dentro do ciclo que percorre os licitadores para enviar notificação ao dono da maior licitação, para os outros participantes é enviado que existe uma nova licitação através do método que envia notificações.

Formas de evitar que o servidor fique bloqueado

Como o objectivo do projecto é impossível sem recurso à programação paralela, esta traz um problema que principalmente é a concorrência.

Nas zonas críticas do nosso código como por exemplo no método envio de notificações que pode ser chamado diversas vezes paralelamente foi usado o recurso do synchronized assim uma thread tem de esperar que a outra acabe a sua execução e só depois é que pode executar este método.

Guardar e ler de ficheiro os plafonds do licitador

O plafond é um atributo do Licitador e em "Guardar e ler de ficheiro as credenciais dos licitadores" quando se deserializa o ficheiro de licitadores na lista de Objetos Licitador o plafond de cada licitador já é guardado em memória.

Guardar e ler de ficheiro os leilões e propostas

Como foi descrito em "Guardar e ler de ficheiro as credenciais dos licitadores" para guardar e ler de ficheiro os leilões o que irá mudar é apenas o nome do ficheiro, as propostas são um atributo da classe Leilão.

Funcionalidade do licitador de fazer pedidos automáticos

Para o gerador de utilizadores que fazem pedidos automáticos inicialmente é pedido ao utilizador se quer pedidos manuais ou automáticos, se a escolha for os automáticos é iniciado o método na classe regulador que autenticar cada um dos clientes que utilizador quiser criar estes clientes são guardados numa lista de utilizadores automáticos. Foi feito um ciclo e neste ciclo para cliente automático é criado uma thread que irá iniciar o método de fazer pedidos automáticos.