



UNIVERSIDADE
LUSÓFONA

Redes de Computadores

Projeto
2018 / 2019

Documento DRAFT

Leiloeira Fanecas

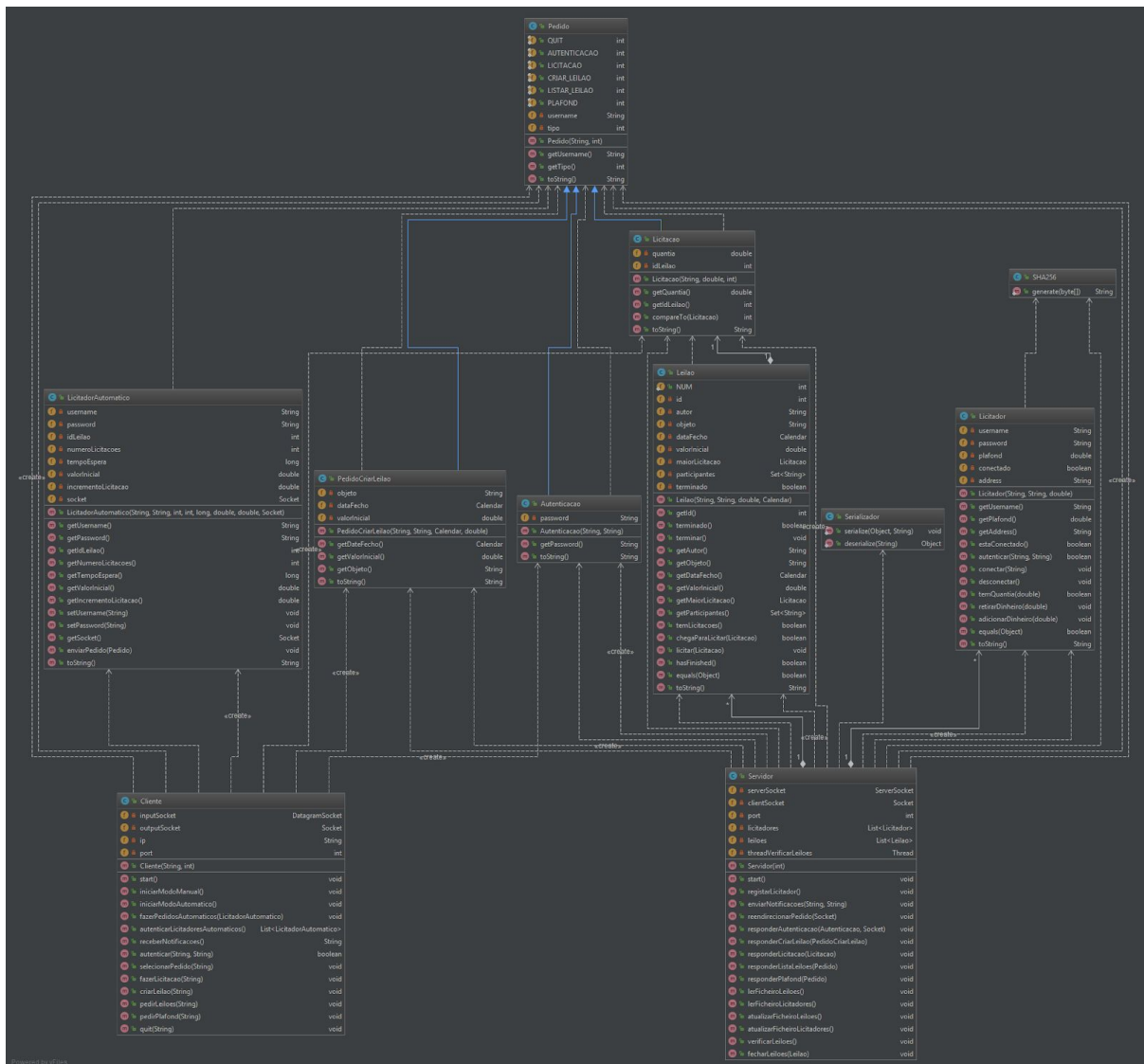
Universidade Lusófona de Humanidades e Tecnologias

Licenciatura em Engenharia Informática

Docentes: Miguel Tavares, José Faísca

Discentes: Francisco Silva 21705328, Rodrigo Cassanheira 21703091

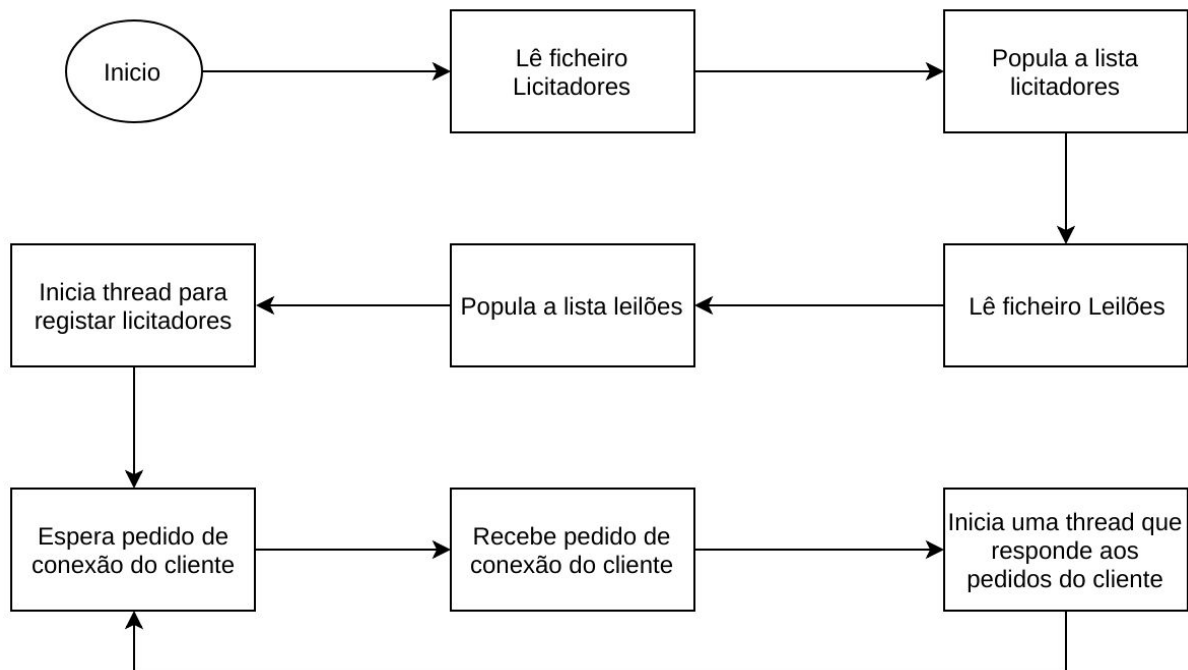
UML



Na figura acima encontra-se o UML do Projeto Leiloeira Fanecas.

Servidor

O servidor ao ser inicializado lê os ficheiros de leilões e licitadores e carrega-os para a sua lista correspondente. Após a leitura dos ficheiros o servidor inicia uma thread que regista licitadores e de seguida começa à espera de conexões. Ao receber uma conexão, o servidor cria uma thread para responder aos pedidos do cliente correspondente. A figura abaixo descreve o processo.



Registar licitadores

Como foi dito anteriormente, para registar licitadores o servidor inicia uma thread. Nesta thread o servidor chama um método num ciclo infinito que pede ao utilizador para inserir o username, se o username existir pede para o inserir novamente, de seguida pede a password, transforma-a num hash, e pede o plafond. Depois o servidor cria um objeto do tipo Licitador e adiciona-o à lista de licitadores. Finalmente, o servidor atualiza a lista de licitadores.

Responder pedidos

O servidor, ao receber uma nova conexão, cria uma thread que chama um método que lê do socket o pedido.

Este método, ao ler o pedido do socket identifica o seu tipo. Isto só é possível uma vez que existem várias classes que herdam da classe Pedido que permitem enviar todos os dados necessários para responder ao pedido enviado pelo cliente.

O pedido ao ser identificado é direcionado para o método que o resolve.

Responder a pedidos

Criar leilão

Após o servidor identificar que o pedido é um pedido de criar licitação, o servidor cria um objeto da classe PedidoCriarLicitação que herda da classe Pedido, e chama um método que responde a este pedido.

Este método recebe o objeto da classe PedidoCriarLeilão e com os seu dados cria um objeto da classe Leilão, adiciona-o à lista de leilões, atualiza o ficheiro de leilões e informa todos os clientes da presença de um novo leilão.

Licitar

O servidor ao identificar que o pedido é uma licitação, o servidor cria um objecto da classe Licitação que herda de Pedido, e chama um método que responde à licitação.

Este método recebe o objecto do tipo Licitação criado. Dentro deste método identifica o leilão em que se foi licitado, faz as validações ao plafond do licitante, caso passe às validações retira o montante licitado e envia uma notificação ao licitante informando que a sua licitação foi aceite. Finalmente envia a todos os participantes do leilão licitado que houve uma licitação que foi feita recentemente.

Pedir plafond

O servidor ao identificar o pedido pedir plafond irá criar um objecto da classe Pedido, e chama um método que responde ao pedido de plafond.

Este método recebe o objecto criado acima, dentro deste método existe um ciclo que identifica o licitador que fez o pedido e irá notificá-lo com o plafond correspondente.

Listar leilões

Uma vez identificado o pedido como um pedido de listar leilões, o servidor cria um objeto da classe Pedido e chama um método para o resolver.

Este método percorre a lista de leilões e para cada leilão envia uma string que o descreve ao cliente que fez o pedido.

Autenticação

Caso o servidor identifique que o pedido é uma autenticação, o servidor cria um objeto do tipo Autenticação e chama um método para o resolver.

Este método, recebe o objeto do tipo Autenticação, transforma a password deste objeto num hash, e procura um licitador cujo username e a password sejam iguais ao username do objeto do tipo Autenticação e ao hash da password deste objeto.

Cliente

O cliente ao iniciar pede ao utilizador que se autentique. De seguida, mostra ao utilizador uma lista de opções. Cada opção corresponde a um tipo de pedido. Após o utilizador escolher uma opção o cliente pede os dados do pedido correspondente, tal como foi referido anteriormente por cada pedido será criado um objeto Pedido ou de outra classe que herda deste e será enviado ao servidor.

Para receber as mensagens o cliente tem uma thread que num ciclo infinito está a receber mensagens do servidor e imprime-as no ecrã.