# PI-5

CELSO VENANCIO LEITE

# Apresentação do projeto

Este projeto tem como objetivo implementar um simulador de defesa antiaérea conforme especificação. Onde haverá uma aplicação cliente-server para simular um radar que detectará e informará a posição exata de um avião em espaço aéreo não autorizado utilizando o protocolo TCP/IP (cliente). Outra aplicação simulará o canhão (servidor) que receberá a posição do avião com algum atraso devido ao meio físico e calculará um ponto onde o avião deverá ser abatido, informando o radar do momento do disparo para que este possa verificar se o avião foi abatido ou não. O objetivo do projeto é a experiência no desenvolvimento de aplicativos com comunicação entre computadores. Como primeira entrega foi solicitado um aplicativo cliente (Radar) e outro servidor (Canhão), em que o servidor ao aceitar a conexão do cliente, envia um pacote contendo algum dado e espera outro de volta. O tempo de *throughput* é calculado. Este processo é repetido por 10 interações e o tempo médio é apresentado ao final.

# Tecnologia utilizada

Ambos aplicativo, cliente e servidor, foram desenvolvidos em *C#* utilizando o *.NET Framework v3.5* no *Visual Studio 2013*.

# Estudo necessário

Somente as bibliotecas básicas (*mscorlib*, *System*) e de rede (*System*.*Net*) do .*NET* *Framework* *v3.5* foram utilizadas. As principais classes utilizadas para comunicação de rede são *TcpClient*, *TcpListener* e *Socket* que são abstrações sobre a camada de comunicação de rede do Windows (*Winsock*). Já para medição do tempo utilizou-se o *Stopwatch* que mede tempo com alta precisão, implementado utilizando a função do Windows *QueryPerformanceFrequency*.

*TcpListener* é utilizado pelo servidor para aguardar pela conexão do cliente. É necessário informar o endereço e porta local para o qual será feito o *bind* além da chamada do método *Start* para que o sistema operacional comece a repassar a comunicação endereçada ao endereço do *bind*. Após isso é chamado o método *AcceptTcpClient* que aguarda pela conexão de algum cliente por tempo indefinido. Quando o cliente, se conectar um objeto do tipo *TcpClient* é retornado permitindo o acesso ao objeto *Socket* para a comunicação de fato. O *TcpClient* também é utilizado pelo cliente para se conectar ao servidor através do método *Connect* que espera o IP e porta de destino. Os métodos *Send* e *Receive* do objeto tipo *Socket são* utilizados para enviar e receber bytes através da rede.

# Resultado obtido

Os prints abaixo mostram a execução do aplicativo Canhão conectado ao aplicativo Radar, em três execuções diferentes onde em cada uma delas é realizado o envio e recebimento de pacotes por dez interações e o tempo médio de cada interação, a texto da última mensagem recebida e o tamanho dessa mensagem são apresentados. O código das duas aplicações pode ser obtido no *Github* no endereço <https://github.com/celsovlpss/BCC-1s16-PI5>.





