**Disciplina:** Redes de Computadores II

**Professora:** Maria de Fatima Webber do Prado Lima

**Alunos:** Gian Carlos Casagrande e Leonardo Bertele Tosin

**Semestre:** 2019/4

**Trabalho Semestral: Analisador de pacotes de rede – Sniffer**

A aplicação foi desenvolvida com base no repositório obtido no site GitHub, link de acesso: <https://github.com/TheLe0/NetworkSnifferLib>, onde o mesmo descreve uma aplicação elaborada na linguagem C#, que obtém pacotes da rede local.

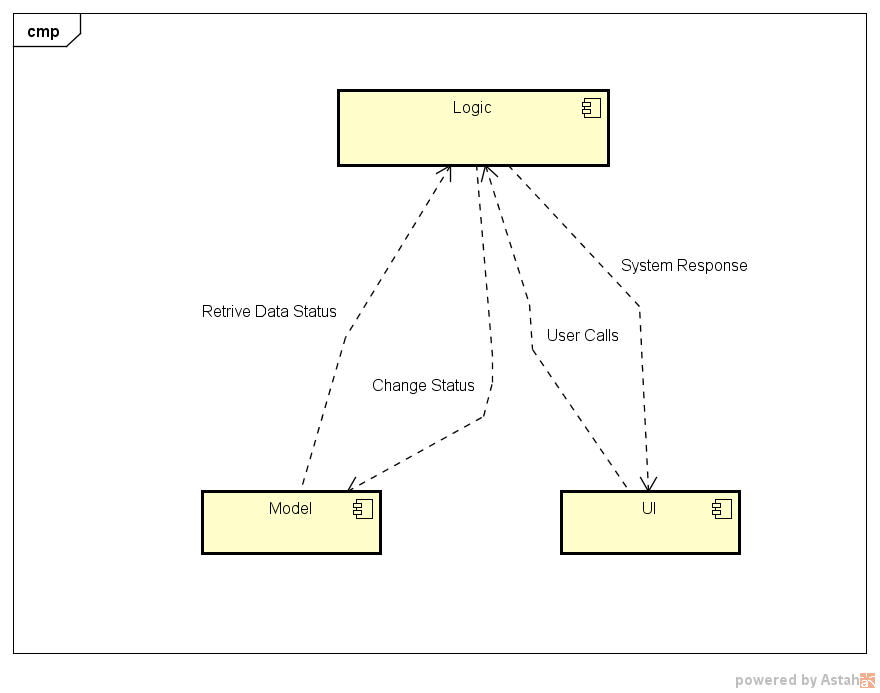
Na nova versão implementada, manteve-se a linguagem C#, e um novo repositório foi criado para a mesma. Algumas alterações adicionais foram feitas, dentre elas estão a análise e processamento das informações obtidas nos pacotes, e a forma de exibição dos dados.

Suas principais funcionalidades são: Obter pacotes presentes no tráfego da rede, analisa-los, e exibir informações dos mesmos, tais como:

* Endereço IP de origem e destino.
* Tipo de protocolo da camada de transporte.
* TTL.
* Número total de pacotes obtidos.
* Número total de pacotes do tipo TCP.
* Número total de pacotes do tipo UDP.
* Número total de pacotes perdidos.
* Informações de tamanho do cabeçalho e de mensagem.

O sistema está organizado em três camadas: Model, UI, e Logic.

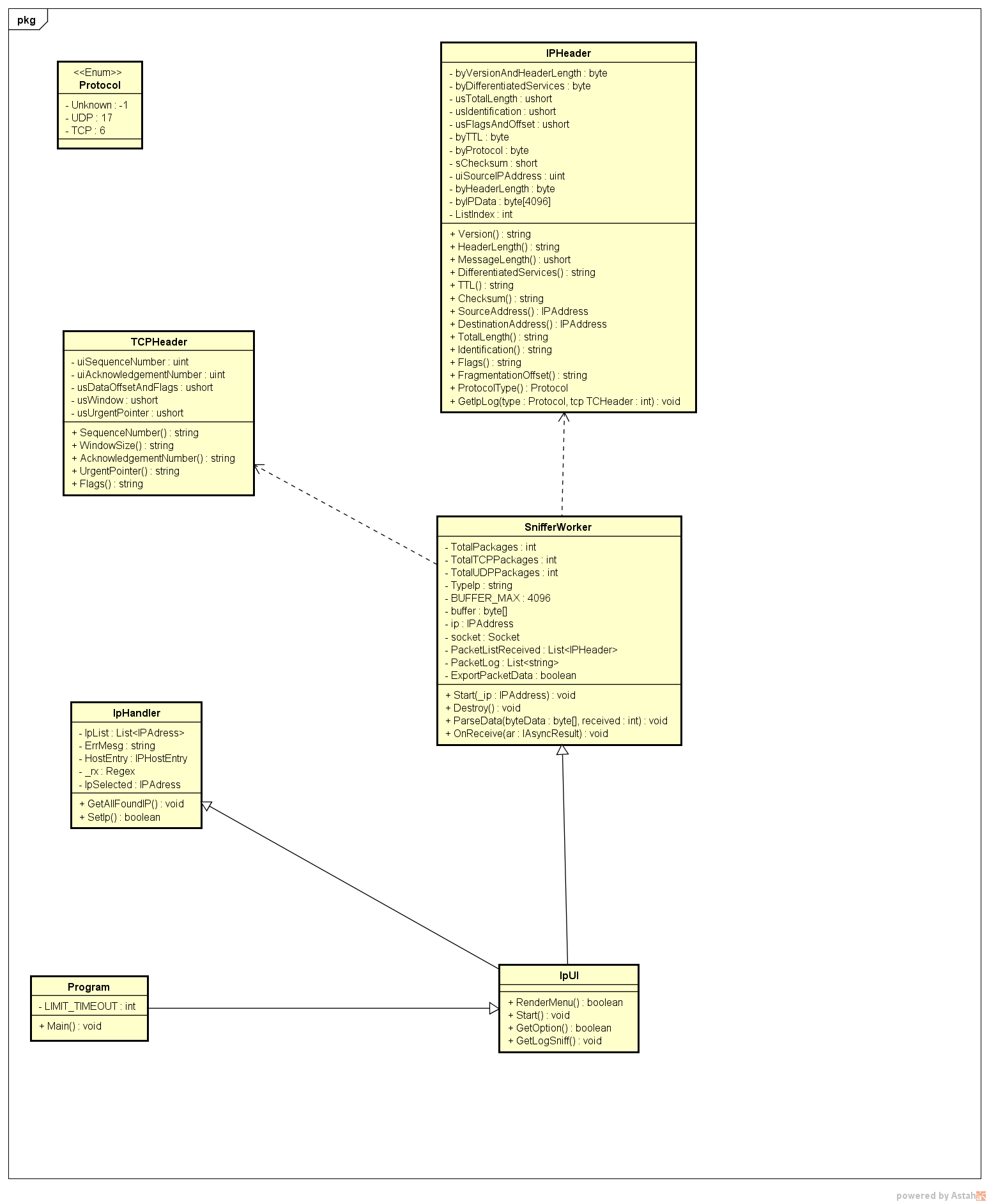
Pode-se observar sua arquitetura no diagrama de componentes abaixo:



As classes presentes em cada componente são:

* **Model:** IPHeader, Protocol, e TCPHeader.
* **UI:** IpUI
* **Logic:** IpHandler, e SnifferWorker.

A divisão das classes pode ser evidenciada no diagrama de classes a seguir:



Funções e seus papéis em cada classe:

Classe Program:

|  |  |
| --- | --- |
| **Função:** | **Descrição:** |
| Main() | Responsável pela chamada dos métodos. |

Classe IpUI:

|  |  |
| --- | --- |
| **Função:** | **Descrição:** |
| RenderMenu() | Lista todos IP´s encontrados no Host, seleciona o IP, e chama a função de obtenção de pacotes. |
| Start() | Chama a função de obtenção de pacotes com o IP selecionado. |
| GetOption() | Lê e retorna o valor lido. A opção de valor selecionada corresponde à posição do IP na lista de IP´s. |
| GetLogSniff() | Informa na tela os dados e informações obtidas durante a análise dos pacotes. |

Classe IpHandler:

|  |  |
| --- | --- |
| **Função:** | **Descrição:** |
| GetAllFoundIP() | Exibe todos os elementos (IP´s) inseridos na lista de endereços IP. |
| SetIp(int ip) | Faz a verificação se o valor de opção (seleção de IP) corresponde às opções de seleção disponíveis, e seta o IP selecionado. |

Classe SnifferWorker:

|  |  |
| --- | --- |
| **Função:** | **Descrição:** |
| Start(IPAddress \_ip) | Faz a obtenção dos pacotes. |
| Destroy() | Fecha o socket se o mesmo não for nulo. |
| ParseData(byte[] byteData, int received) | Processa, analisa os pacotes recebidos, e armazena as informações em uma lista. |
| OnReceive(IAsyncResult ar) | Destrava os ponteiros. |

Classe TCPHeader:

|  |  |
| --- | --- |
| **Função:** | **Descrição:** |
| SequenceNumber() | Retorna o número de sequência. |
| WindowSize() | Retorna o window size. |
| AcknowledgementNumber() | Retorna o número de reconhecimento. |
| UrgentPointer() | Retorna o Urgent Pointer. |
| Flags() | Retorna as flags. |

Classe IPHeader:

|  |  |
| --- | --- |
| **Função:** | **Descrição:** |
| Version() | Retorna a versão do IP. |
| HeaderLength() | Retorna o tamanho do cabeçalho. |
| MessageLength() | Retorna o tamanho da mensagem. |
| DifferentiatedServices() | Retorna os serviços diferenciados da rede. |
| TTL() | Retorna o valor do TTL. |
| Checksum() | Retorna o valor do Checksum. |
| SourceAddress() | Retorna o endereço fonte. |
| DestinationAddress() | Retorna o endereço de destino. |
| TotalLength() | Retorna o comprimento total. |
| Identification() | Retorna a identificação. |
| Flags() | Retorna as flags. |
| FragmentationOffset() | Retorna o deslocamento de fragmentação. |
| ProtocolType() | Retorna o tipo de protocolo. |
| GetIpLog(Protocol type, TCPHeader tcp) | Retorna informações do log obtidas no IP. |