

Victor Correia de Moura - 15/0150792

Vitor Falcão H. Costa - 15/0151624

Introdução

As dimensões territoriais do Brasil e a sua diversificação trazem grandes vantagens ao povo brasileiro e sua cultura. Entretanto, é um verdadeiro desafio adquirir e distribuir recursos que atingem grande parte da população que não está presente em grandes centros urbanos. A gestão da saúde torna-se a cada dia mais inviável, já que grandes polos de conhecimento não conseguem prover serviços a locais menos urbanizados.

Objetivos

O objetivo do projeto em questão é criar um sistema que pode viabilizar a realização de exames de ultrassom operados à distância por um médico especialista. Assim, será possível um paciente consultar-se com um grande conhecedor da área de conhecimento sem que seja necessário deslocar-se fisicamente ao consultório.

Tecnologias existentes no mercado

Atualmente, existem soluções que aproveitam o conceito de telemedicina para conseguir descentralizar o atendimento médico com especialistas de diversas áreas da medicina. Entretanto, a grande maioria se baseia na ideia de se enviar os resultados, em tempo real ou não, a um especialista.

O diferencial do projeto seria a possibilidade de se operar o equipamento de ultrassom à distância por meio de um joystick ou alguma outra interface. Sabe-se que isso é possível pela existência de procedimentos cirúrgicos que fazem uso desse princípio.

Desenvolvimento

A arquitetura inicial que poderia tornar viável o desenvolvimento é o uso de um servidor TCP/IP em python para realizar o streaming de dados de um ponto a outro via internet. Esses dados podem ser enviados para o computador cliente e os dados são codificados em informação visível na tela. O protocolo TCP/IP já define os *handshakes* necessários para se manter a integridade da informação. Ainda não se definiu se o servidor será um dos computadores que vão estar no local de exame e no local onde o especialista se encontra, ou se será uma terceira máquina que realizará a comunicação de outras. A dificuldade da implementação pode ser maior mas facilitaria a escalabilidade do produto.

Da mesma maneira que o servidor pode ser em Python, os *clients* podem se aproveitar da mesma tecnologia e oferecer o recebimento e envio de dados ao servidor. Os dados recebidos podem ser transmitidos para alguma interface de controle do braço mecânico, possibilitando a sua movimentação. Os dados de vídeo podem ser mostrados diretamente na tela do *client*. O *client* pode ser desenvolvido em Python, também, fazendo o uso do pacote TkInter para possibilitar o uso de interface gráfica para o usuário.

A latência da transmissão é dependente de dois fatores: processamento interno dos dados e infraestrutura de internet. O cálculo da banda necessária para a transmissão dos dados depende dos equipamentos selecionados pelos grupos responsáveis pelo *hardware* do produto.

Conclusão

A arquitetura proposta possui o potencial necessário para o envio e recebimento de dados entre dois pontos conectados à internet. Esses dados podem ser quaisquer informações que possam ser traduzidas em bits. Tendo isso em mente, há evidências de que é possível a sua implementação para que o processo da comunicação entre o equipamento do médico especialista e o braço mecânico possa ser realizado.

Cabe ressaltar que cabe estudo da viabilidade do uso de requisições HTML para se realizar a implementação. A existência de frameworks facilitaria o desenvolvimento mas não se sabe se haveria problema de processamento de dados se utilizada essa abordagem.

Referências

- [1] Cirurgiões realizam cirurgias a distância no Canadá. UNA-SUS <https://www.unasus.gov.br/noticia/cirurgioes-realizam-cirurgias-distancia-no-canada>
- [2] Telemedicina. Einstein. <https://www.einstein.br/empresas-hospitais/produtos-servicos/telemedicina>
- [3] Institucional. Brasil Telemedicina. <http://brasiltelemedicina.com.br/institucional/>
- [4] TCP/IP Client and Server. PyMOTW. <https://pymotw.com/2/socket/tcp.html>
- [5] Transmission Control Protocol. Wikipedia. https://en.wikipedia.org/wiki/Transmission_Control_Protocol
- [6] TkInter. Python. <https://wiki.python.org/moin/TkInter>