# Teste

Quando se corre o programa o utilizador depara-se com a interface de visualização. Este pressiona o botão de carregamento de imagens e seleciona as imagens a utilizar.

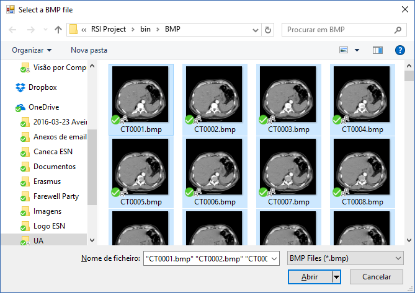


Figura 1- Interface de carregamento de imagens

O utilizador consegue observar o corte transversal à esquerda, o corte coronal no centro e o corte sagital à direita.

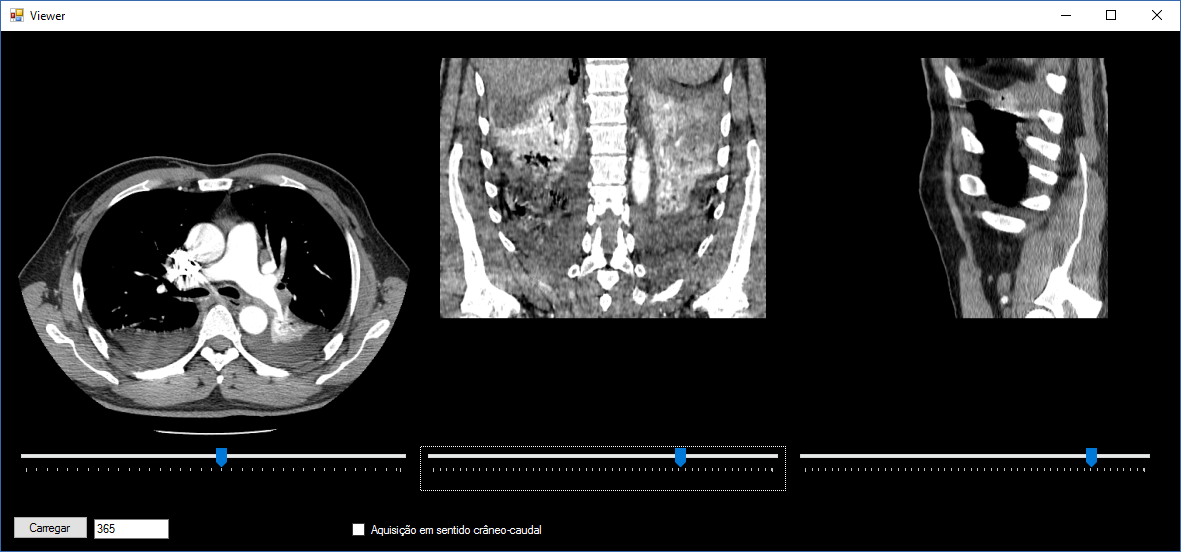


Figura 2- Interface de visualização dos três cortes

Selecionando um ponto na caixa de imagem à esquerda, é possível observar os cortes para o mesmo ponto selecionado nas imagens ao centro e à direita.

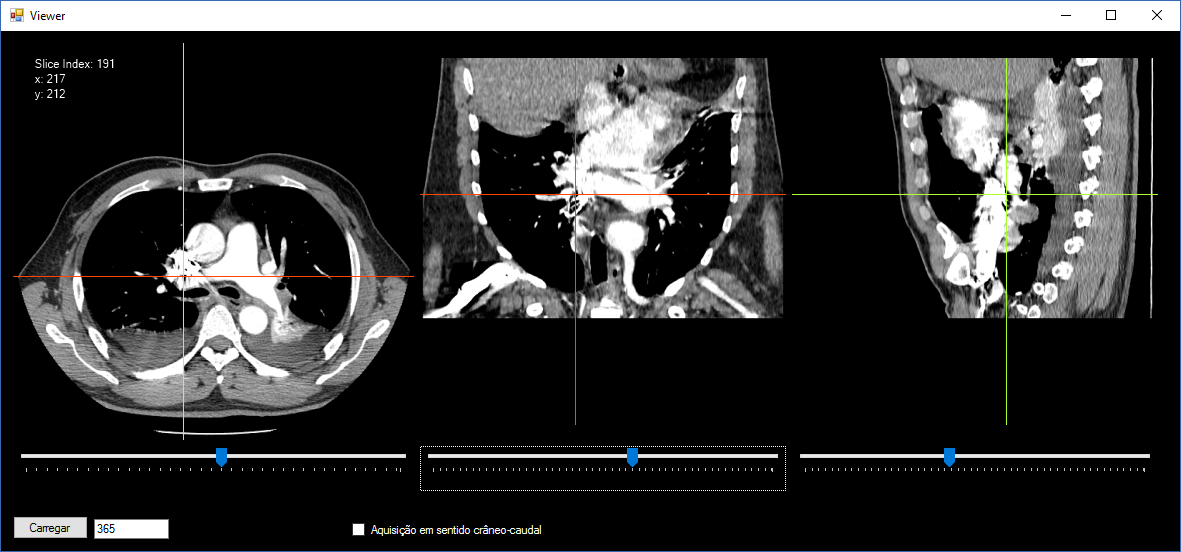


Figura 3- Interface de visualização dos três cortes através do ponto selecionado no plano transversal

# Especificações

O objetivo deste projeto é desenvolver um visualizador que permita a reconstrução seccional de imagens médicas (TAC, RM) em diversos planos, tais como o longitudinal, sagital e transversal. Ao utilizador é-lhe permitido carregar a sequência de imagens no visualizador e controlar a imagem em exibição.

# Algoritmo

O algoritmo base desenvolvido foi pensado de acordo com o seguinte pseudocódigo:

Início

Carregar imagens numa List

Verificar trackbar scroll\_transversal

Verificar trackbar scroll\_coronal

Verificar trackbar scroll\_sagital

Visualizar imagens no Displayer transversal

Carregar as colunas de pixéis de todas as imagens

Visualizar imagens no Displayer coronal

Carregar as linhas de pixeis de todas as imagens

Visualizar imagens no Displayer sagital

# Implementação

O utilizador pressiona no botão para carregar as imagens em formato BMP e de seguida o programa adiciona cada imagem a um elemento da lista *orig\_img*.

foreach (string file in openFileDialog1.FileNames)

{

//Original Images

orig\_img.Add(new Bitmap(file));

int x\_max = orig\_img.Count();

int y\_max = x\_max;

height = orig\_img[0].Height;

width = orig\_img[0].Width;

seedFrontal = new Bitmap(width, height);

seedLateral = new Bitmap(width, height);

}

O utilizador navega entre as imagens de corte transversal através de uma barra de deslizamento *myTB1*, em que cada posição da barra corresponde a uma imagem da lista *orig\_img.* Neste caso as imagens também são carregadas na caixa da imagem.

Para as restantes barras o raciocínio é semelhante.

private void trackBar1\_Scroll(object sender, EventArgs e)

{

System.Windows.Forms.TrackBar myTB1;

myTB1 = (System.Windows.Forms.TrackBar)sender;

currBMP = myTB1.Value;

pictureBox1.Image = orig\_img.ElementAtOrDefault(myTB1.Value);

pictureBox1.Invalidate();

}

Para processar as imagens do plano coronal, carregamos como argumento da função *process\_frontal*, a posição vertical das coordenadas selecionadas. Dependendo agora se a checkbox na interface estava selecionada ou não, mostraremos a sequência de cortes coronais invertida ou não. O ciclo *for* percorre todas as imagens na lista *orig\_img,* vai buscar as linhas de pixéis de cada uma e constrói um novo Bitmap chamada *seedFrontal*.

A mesma lógica é aplicada para o processamento do plano sagital, no entanto usa-se a posição horizontal das coordenadas selecionadas.

private void process\_frontal(int pos)

{

Color newPixel;

if (checkBox1.Checked)

{

int flip = 0;

for (int i = orig\_img.Count() - 1; i >= 0 && flip < orig\_img.Count(); i--)

{

for (int j = 0; j < width; j++)

{

newPixel = orig\_img[i].GetPixel(j, pos);

seedFrontal.SetPixel(j, flip, newPixel);

}

flip++;

}

}

else

{

for(int i = 0; i < orig\_img.Count() - 1;i++)

{

for(int j = 0; j < width; j++)

{

newPixel = orig\_img[i].GetPixel(j, pos);

seedFrontal.SetPixel(j, i, newPixel);

}

}

}

}

Este segmento de código permite selecionar o ponto de corte para os outros planos e desenhar as linhas sobre a caixa da imagem com o plano transversal.

private void pictureBox1\_Paint(object sender, PaintEventArgs e)

{

if (!paintHandler)

{

return;

}

Pen penG = new Pen(Color.GreenYellow, 1);

Pen penR = new Pen(Color.OrangeRed, 1);

e.Graphics.DrawLine(penG, new Point(coordinates.X, 0), new Point(coordinates.X, pictureBox1.Height));

e.Graphics.DrawLine(penR, new Point(0, coordinates.Y), new Point(pictureBox1.Width, coordinates.Y));

int y = height - py;

int idx = currBMP + 1;

e.Graphics.TextRenderingHint = System.Drawing.Text.TextRenderingHint.AntiAlias;

e.Graphics.DrawString("Slice Index: " + idx, Font, Brushes.White, 20, 15);

e.Graphics.DrawString("x: " + px, Font, Brushes.White, 20, 30);

e.Graphics.DrawString("y: " + y, Font, Brushes.White, 20, 45);

}