INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO - CAMPUS CAPIVARI

CURSO: ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

DISCIPLINA: LINGUAGEM DE PROGRAMAÇÃO I

PROFESSOR: CARLOS ROBERTO PAVIOTTI

**Sistema de Travessia de Pontes The Beatles**

Matheus de Barros Fagionato

Emerson Almeida Souza

**Sistema de Travessia de Pontes The Beatles**

Trabalho apresentado como requisito de avaliação na disciplina Linguagem de Programação I do Curso Superior Análise e Desenvolvimento de Sistemas do IFSP - Campus Capivari.

**RESUMO**

Baseado em um problema proposto pelo professor Carlos Roberto Paviotti, onde será calculado o tempo de travessia da banda The Beatles pela ponte Rickety. A banda precisa passar pela ponte para chegar no local da realização de um show. A travessia será durante a noite, e o grupo possui apenas uma lanterna para os quatro integrantes. A travessia será realizada em um determinado tempo definido pelo usuário, onde só poderá atravessar no máximo dois membros dos Beatles. Obrigatoriamente um dos integrantes que está atravessando carregará a lanterna. Todos os Beatles deverão atravessar a ponte no tempo estipulado pelo usuário, para que o show seja realizado.

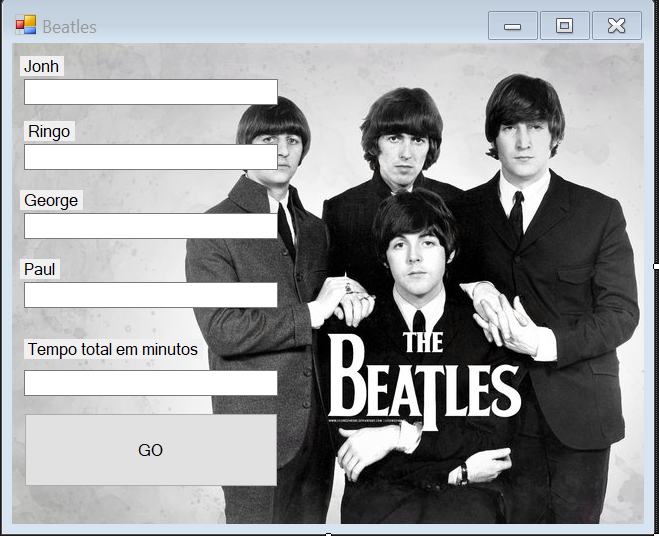
**INTRODUÇÃO**

O ser humano em seu dia-a-dia, se depara com um excesso de atividades rotineiras (trabalhar, estudar etc). Devido a esta demanda de afazeres, ocorre um problema que é comum a todo ser humano, os atrasos. E como todo problema, nos tempos atuais, nos perguntamos se o computador não poderia resolve-lo para nós. E como na maioria das vezes, sim ele pode.

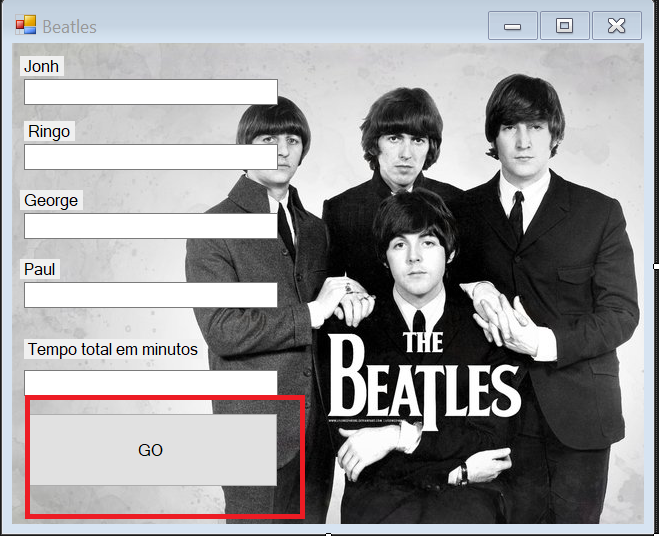
Esta aplicação tem como intuito fazer o planejamento automático para saber se dará tempo ou não de chegar em um determinado local, em um determinado período de tempo usando como exemplo a banda The Beatles. Foi utilizada a linguagem de programação C#, com a ferramenta Visual Studio 2012, para a codificação do algoritmo com a resolução do problema.

**ANALISE DO CÓDIGO**

O sistema de travessia de pontes The Beatles possui duas funcionalidades, nas quais são: Verificar se dará para atravessar a ponte com o tempo fornecido pelo usuário para cada um dos integrantes, e mostrar como eles irão passar

Interface do Sistema de travessia de pontes The Beatles

Ao selecionar o botão “GO”, o programa irá verificar se será possível realizar a travessia a tempo.

Primeira funcionalidade do sistema

Ao selecionar o botão “GO”, o sistema irá verificar se os campos estão vazios com o tratamento de exceção. Se estiver vazio, chamo meu método “limpar”. Onde será solicitado ao usuário que digite os tempos novamente.

**Codificação do método “limpar”:**

public void limpar()

{

tb\_tempo\_george.Text = "";

tb\_tempo\_jonh.Text = "";

tb\_tempo\_paul.Text = "";

tb\_tempo\_ringo.Text = "";

tb\_tempo\_jonh.Focus();//Primeiro campo do formulário

}

Codificação do tratamento de exceção:

private void btn\_Go\_Click(object sender, EventArgs e)

{

try

{

TemposRepitidos(); /\*Verificar se tem campo com valores iguais\*/

PassarPonteComMaisRapido();

}

catch (FormatException)

{

MessageBox.Show("Todos os campos deverão ser preenchidos, digite novamente os tempos");

limpar();

}

}

Logo em seguida o programa irá verificar se contém campos com valores repetidos, pois não poderá existir integrantes que atravessam a ponte ao mesmo tempo.

Codificação para verificar se existem valores repetidos nos campos de texto:

public void TemposRepitidos()/\*Método para caso algum tempo seja igual ao outro, assim daria conflito em quem seria o mais rápido\*/

{

TempoRingo = Convert.ToInt16(tb\_tempo\_ringo.Text);

TempoPaul = Convert.ToInt16(tb\_tempo\_paul.Text);

TempoGeorge = Convert.ToInt16(tb\_tempo\_george.Text);

TempoJonh = Convert.ToInt16(tb\_tempo\_jonh.Text);

if ((TempoRingo == TempoPaul) || (TempoRingo == TempoGeorge) || (TempoRingo == TempoJonh))//Inicio da verificação se tem campos com tempos iguais

{

campos\_repetidos++;

}

if ((TempoPaul == TempoRingo) || (TempoPaul == TempoGeorge) || (TempoPaul == TempoJonh))

{

campos\_repetidos++;

}

if ((TempoGeorge == TempoPaul) || (TempoGeorge == TempoRingo) || (TempoGeorge == TempoJonh))

{

campos\_repetidos++;

}

if ((TempoJonh == TempoPaul) || (TempoJonh == TempoGeorge) || (TempoJonh == TempoRingo))//Fim da verificação de tempos iguais

{

campos\_repetidos++;

}

int TempoTotal = Convert.ToInt16(tb\_tempo\_total.Text);//Tempo total de 17 minutos

if (campos\_repetidos > 0)/\*Se tiver algum campo repetido\*/

{

MessageBox.Show("PREENCHA TODOS OS CAMPOS COM VELOCIDADES DIFERENTES!");

}/\*Onde irá dar erro acaso o usuário digite um campo igual\*/

}

Logo em seguida o programa irá executar a função onde irá verificar se será possível passar com o tempo fornecido pelo usuário, e calcular, partindo do mais rápido, como será realizado a travessia da maneira mais eficiente possível. Pegando sempre o “Beatle” mais rápido, para realizar a travessia com o “Beatle” mais rápido subsequente.

Codificação para verificar se passou e também o “Beatle” mais rápido:

public void PassarPonteComMaisRapido()

{

TempoRingo = Convert.ToInt16(tb\_tempo\_ringo.Text);

TempoPaul = Convert.ToInt16(tb\_tempo\_paul.Text);

TempoGeorge = Convert.ToInt16(tb\_tempo\_george.Text);

TempoJonh = Convert.ToInt16(tb\_tempo\_jonh.Text);

int TempoTotal = Convert.ToInt16(tb\_tempo\_total.Text);//Tempo total de 17 minutos

int[] HereComesTheSun = new int[] {TempoRingo, TempoPaul, TempoGeorge, TempoJonh};

int[] YellowSubarine = HereComesTheSun;

int total = 0;

bool passou;

string BeattleMaisRapido = "";

string BeattleMaisRapido2 = "";

string BeattleMaisRapido3 = "";

string BeattleMaisRapido4 = "";

Array.Sort(HereComesTheSun);/\*Ordenar array de forma crescente\*/

for (int i = 0; i < 4; i++)//Somar o tempo total

{

if(i == 0) { total = HereComesTheSun[i] \* 5; }/\*Quem tá na posição 0 sempre será o mais rápido\*/

if(i == 1) { total += HereComesTheSun[i]; }

if(i == 2) { total += HereComesTheSun[i]; }

if(i == 3) { total += HereComesTheSun[i]; }

}

int tempoTotal = TempoRingo + TempoPaul + TempoGeorge + TempoJonh;

if (TempoTotal >= total)//Se o tempo de todos juntos for menor que o tempo total

{

;/\*Calculo do tempo total para passagem\*/

MessageBox.Show("Os beatles estão atrevessando a ponte Rickety. Irão demorar"+" "+tempoTotal+" minutos para completar a travessia");

passou = true;

}

else

{

MessageBox.Show("ESTÃO MUITO DEVAGAR, NÃO CONSEGUIRÃO PASSAR A PONTE EM " +tempoTotal+" MINUTOS"+"..\n\n"

+"TENTE IR MAIS RÁPIDO DESTA VEZ!!!"

);

passou = false;

limpar();//Quando não conseguir passar eu limpo tudo

}

if (passou == true)/\*Se passaram, organiza como passarão\*/

{

if (TempoRingo == HereComesTheSun[0]){

BeattleMaisRapido = "RINGO";

BeattleMaisRapido2 = "GEORGE";

BeattleMaisRapido3 = "PAUL";

BeattleMaisRapido4 = "JOHN";

}

if (TempoGeorge == HereComesTheSun[0]){

BeattleMaisRapido = "GEORGE";

BeattleMaisRapido2 = "RINGO";

BeattleMaisRapido3 = "PAUL";

BeattleMaisRapido4 = "JOHN";

}

if (TempoJonh == HereComesTheSun[0]){

BeattleMaisRapido = "JOHN";

BeattleMaisRapido2 = "GEORGE";

BeattleMaisRapido3 = "PAUL";

BeattleMaisRapido4 = "RINGO";

}

if (TempoPaul == HereComesTheSun[0]){

BeattleMaisRapido = "PAUL";

BeattleMaisRapido2 = "GEORGE";

BeattleMaisRapido3 = "RINGO";

BeattleMaisRapido4 = "JOHN";

}

//Passando a ponte

MessageBox.Show("PASSA " + BeattleMaisRapido + " (Com a Lanterna) E " + BeattleMaisRapido2);///Sempre o primeiro mais rápido irá passar

MessageBox.Show("VOLTA " + BeattleMaisRapido + " (Com a Lanterna) ");

MessageBox.Show("PASSA " + BeattleMaisRapido + " (Com a Lanterna) E " + BeattleMaisRapido3);

MessageBox.Show("VOLTA " + BeattleMaisRapido + " (Com a Lanterna)");

MessageBox.Show("PASSA " + BeattleMaisRapido + " (Com a Lanterna) E " + BeattleMaisRapido4);

MessageBox.Show("TODOS OS BEATLES CONSIGUERAM CHEGAR NO SHOW A TEMPO!!!!");

limpar();

}

}

Se for possível passar, o sistema exibira o passo a passo para a travessia da Banda, assim como o tempo utilizado para completar o trajeto. Se não, o programa irá exibir uma mensagem solicitando ao usuário que tente novamente com menos tempo, exibindo o tempo que ele utilizou para tentar a trevessia.

**CONCLUSÃO**

O trabalho realizado para desenvolver um sistema de travessia de pontes para a banda The Beatles, deu uma noção de como um programador trabalha em seu dia-a-dia. Dividindo as tarefas entre a equipe e trabalhando em conjunto. Para que assim fosse desenvolvido da melhor forma possível. O trabalho também trouxe uma visão de como seria um sistema criado com base em exigências de um cliente. Onde foi estipulado um prazo de entrega, e requisitos para o sistema. Este trabalho nos deu uma pequena visão de como seria o mercado de trabalho para o programador, onde diariamente os clientes surgem com problemas, onde nos enquanto desenvolvedores com nossos conhecimentos trabalhamos em conjunto com a nossa equipe, para desenvolver uma solução em software para o cliente.

**REFERÊNCIAS BIBLIOGRÁFICAS**

*Canal no YouTube do João Ribeiro.* [*C# - De Novato a Profissional*](https://www.youtube.com/playlist?list=PLXik_5Br-zO-rMqpRy5qPG2SLNimKmVCO)*.* Disponível em: < https://www.youtube.com/watch?v=IMA5Jr3iMGI&list=PLXik\_5Br-zO-rMqpRy5qPG2SLNimKmVCO >. Acesso em: 24 nov. 2019.