**TRY CATCH FINALLY**

**O QUE É:**O try catch é um bloco de comando que funciona por etapas e é muito parecido com um IF. Ele tem a função de lidar com erros de forma controlada.  
  
**ETAPAS:**

* **TRY –** Esta parte contém o código do usuário que deseja monitorar os possíveis   
  Texto

  Descrição gerada automaticamente
* **CATCH -** Aqui é onde você define filtros podem ser lançadas no bloco **TRY** e pode ter múltiplos blocos catch para capturar diferentes tipos de exceções.   
  Texto

  Descrição gerada automaticamente
* **FINALLY**: Este bloco é opcional e contém o código que será executado independentemente de uma exceção ter sido lançada ou não. É útil para limpar recursos, como fechar arquivos ou conexões de rede, que precisam ser liberados independentemente de um erro ter ocorrido.

Texto

Descrição gerada automaticamente

**CÓDIGO PARA FORÇAR EXECUÇÃO DO TRY CATCH FINALLY**Um código bem simples para executar os blocos TRY CATCH FINALLY exemplificado:

namespace TryCatch

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private void ForceTryCatch()

{

try

{

//Código que pode lançar um erro proposital

int[] array = new int[5];

Console.WriteLine(array[10]); //Isso resultará em um erro pois está fora dos limites do Array

}

catch (IndexOutOfRangeException ex)

{

//Catch para lidar com o erro

MessageBox.Show("O Vetor do array não existe: " + ex.Message);

}

finally

{

//O Finally que será executado sempre, independentemente de houver erros ou não

MessageBox.Show("Finally executado.");

}

}

private void btnTeste\_Click(object sender, EventArgs e)

{

ForceTryCatch();

}

}

}

Nele, iniciamos a execução de um **TRY** que lança um erro proposital; gerar a variável **array** = [5] e chamar o vetor [10] (que excede o que foi definido na variável **array**) causa o erro.

Em seguida, capturamos o erro com o **CATCH** e informamos uma mensagem para o usuário juntamento do código de erro.

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Por fim, executamos o **FINALLY** que mostrará uma mensagem para o usuário, independentemente se houve erros no **TRY.**

Interface gráfica do usuário, Texto, Aplicativo, chat ou mensagem de texto

Descrição gerada automaticamente

A janela GUI do sistema se dá por:

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente