



**NICOLLAS CAVALCANTE SOUZA**

**PROGRAMA: Quiz Conhecimento sobre Macacos**

**MOGI DAS CRUZES 2023**  
**NICOLLAS CAVALCANTE SOUZA**  
**RGM 11221104304**

## **RESUMO**

Neste Trabalho temos como objetivo a melhor construção do aprendizado de alguns conceitos de programação orientada a objeto como, variáveis globais, ocultação de campos e forms e passagem de informação entre um e o outro, integração com Banco de dados utilizando Classes DAO aplicando o conceito de Data Acces Object. Para isto foi feito um Quiz de tema livre o escolhido por mim foi sobre curiosidades de primatas.

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>0</b>
<b>2</b>	<b>DESCRIÇÃO DOS RECURSOS UTILIZADOS.....</b>	<b>06</b>
<b>2.1</b>	<b>Descrição dos Atributos Utilizados Detalhados .....</b>	<b>06</b>
<b>2.2</b>	<b><i>Funcionamento no Banco de dados</i> .....</b>	<b>10</b>
<b>3</b>	<b>Código .....</b>	<b>11</b>
<b>4</b>	<b>Funcionamento .....</b>	<b>16</b>
<b>5</b>	<b>Conclusão.....</b>	<b>21</b>

## 1 INTRODUÇÃO

Neste trabalho realizei um estudo sobre os Quiz e seus formatos de código e logicas para programação e me baseando em alguns exemplos vistos internet afora e os que o professor passou em sala, utilizando a Linguagem de programação C# e o simulador de banco de dados locais XAMPP criei um aplicativo onde poderíamos realizar um questionário com 3 alternativas por perguntas, guardando em Banco de dados o total de erros e acertos do jogador, assim como seu Nome e seu RGM, sendo possível se consultar no Menu os resultados .

A fim de conhecimento neste trabalho utilizamos 5 Classes no total

Também foi utilizado no total 2 tabelas no Banco de dados: Resposta, Questoes  
Com um total de 8 forms, onde 1 é o Menu, 5 são Questões, 1 é o Final e preenchimento de RGM e Nome, e um outro é para a vistoria dos resultados de todos.

## 2 Descrição de Recursos Utilizados

Ao decorrer da criação utilizamos alguns recursos eles são:

Programa para criação de Banco de dados: XAMPP

Programa para criação do Código: Visual Studio 2019

Linguagem de programação Utilizada: C#

### 2.1 Descrição de atributo utilizados detalhada

Para a criação do programa Utilizei um total de 5 Classes:

1. Questoes
2. Respostas
3. QuestoesDAO
4. ConectaDB
5. VariaveisGlobais

Onde Cada uma continha alguns atributos específicos e seus devidos métodos Get/Set e métodos construtores e nos casos das outras classes, métodos específicos

#### Questoes:

```
public class Questoes
{
    //Atributos
    private int ID;
    private String Descricao;
    private String A1;
    private String A2;
    private String A3;
    private String Resposta;

    //Classe Bob(Constructor)

    0 referências
    public Questoes(int p_id, String p_descricao, String p_A1, String p_A2, String p_A3, String p_Resposta)
    {
        this.ID = p_id;
        this.Descricao = p_descricao;
        this.A1 = p_A1;
        this.A2 = p_A2;
        this.A3 = p_A3;
        this.Resposta = p_Resposta;
    }

    //The real Bob Construtor

    5 referências
    public Questoes ()
    {

```

## Respostas:

```

using System;
19 referências
public class Respostas
{
    //Atributos
    private String Nome;
    private String RGM;
    private int TotalAcerto;
    private int TotalErro;

    //Bob Construtor

    //Construtor com todas infos
    0 referências
    public Respostas (String p_Nome, String p_RGM, int p_TotalAcerto, int p_TotalErro)
    {
        this.Nome = p_Nome;
        this.RGM = p_RGM;
        this.TotalAcerto = p_TotalAcerto;
        this.TotalErro = p_TotalErro;
    }

    //The real default Bob Construtor
    8 referências
    public Respostas ()
    {
    }
}

```

A partir daqui todas classes são para controles

## QuestõesDAO:

Esta Classe utiliza o Conceito do Data Acces Object, e possui métodos para conectar ao banco de dados e trazer informações ou cadastrar, exemplo: Cadastrar, Consultar...

```

10 referências
public class QuestoesDAO
{
    //Conexão para questões
    5 referências
    public Questoes consultarId(Questoes quest)...

    //Conexão para Respostas

    //Cadastrar
    1 referência
    public bool Cadastrar(Respostas resp)...

    //Puxar para visualizar

    1 referência
    public DataTable consultarRespostas()...
}

```

Os 3 métodos utilizados foram os seguintes:

### - Consultar ID

Neste método foi realizado a conexão ao DB e fizemos um Select onde o ID é igual ao que desejamos para trazer as questões ao sistema

```
public Questoes consultarId(Questoes quest)
{
    String sql_SELECT = @"SELECT * FROM questões WHERE ID = ?";

    //Database
    try
    {
        MySqlConnection conexao = ConectaDB.conectar();
        conexao.Open();
        MySqlCommand sql_comando = new MySqlCommand(sql_SELECT, conexao);
        sql_comando.Parameters.AddWithValue("@ID", quest.GetID());

        MySqlDataReader dataReader = sql_comando.ExecuteReader();
        dataReader.Read();

        // Atualiza - set
        quest.SetDescricao(dataReader.GetValue(1).ToString());
        quest.SetA1(dataReader.GetValue(2).ToString());
        quest.SetA2(dataReader.GetValue(3).ToString());
        quest.SetA3(dataReader.GetValue(4).ToString());
        quest.SetResposta(dataReader.GetValue(5).ToString());
        conexao.Close();
    }
    catch (Exception exc)
    {
        MessageBox.Show("Erro: " + exc);
    }
    return quest;
}
```

### - Cadastrar

Neste acessamos o banco de dados na tabela “Respostas” para cadastrar as informações do jogador e seus acertos/erros

```
//Cadastrar
1 referência
public bool Cadastrar(Respostas resp)
{
    //Cadastrar - Insert
    string sql_INSERT = @"INSERT INTO resposta (Nome, RGM, TotalAcerto, TotalErro) VALUES(?, ?, ?, ?)";

    //Database
    try
    {
        MySqlConnection conexao = ConectaDB.conectar();
        conexao.Open();
        MySqlCommand sql_comando = new MySqlCommand(sql_INSERT, conexao);

        sql_comando.Parameters.AddWithValue("@Nome", resp.GetNome());
        sql_comando.Parameters.AddWithValue("@RGM", resp.GetRGM());
        sql_comando.Parameters.AddWithValue("@TotalAcerto", resp.GetTotalAcerto());
        sql_comando.Parameters.AddWithValue("@TotalErro", resp.GetTotalErro());

        sql_comando.ExecuteNonQuery();
        conexao.Close();
        return true;
    }
    catch (Exception exc)
    {
        MessageBox.Show("Erro: " + exc);
        return false;
    }
}
```

### - ConsultarRespostas

Neste também acessamos a respostas para pegar um “Placar” por assim dizer de resultados

```
//Puxar para visualizar

1 referência
public DataTable consultarRespostas()
{
    DataTable dt = new DataTable();
    String sql_SELECT = @"SELECT * FROM resposta";

    //Database
    try
    {
        MySqlConnection conexao = ConectaDB.conectar();
        conexao.Open();

        //Ajustar os dados
        MySqlDataAdapter da = new MySqlDataAdapter(sql_SELECT, conexao);

        using (da)
        {
            da.Fill(dt);
        }
        conexao.Close();
    }
    catch (Exception exc)
    {
        MessageBox.Show("Erro: " + exc);
    }
    return dt;
}
```

### ConectaDB:

Aqui foi feito uma classe justamente para conectar ao DB, ela foi utilizada muitas vezes na classe “QuestoesDAO”

```
1 using MySql.Data.MySqlClient;
2
3 3 referências
4 public class ConectaDB
5 {
6     3 referências
7     public static MySqlConnection conectar()
8     {
9         // Banco de Dados
10        return new MySqlConnection("server=localhost; database=quiz; user id=root; password='');
11    }
12 }
```

### VariaveisGlobais:

Aqui foi criado duas variáveis globais para poder somar em todos forms a quantia de erros e acertos.

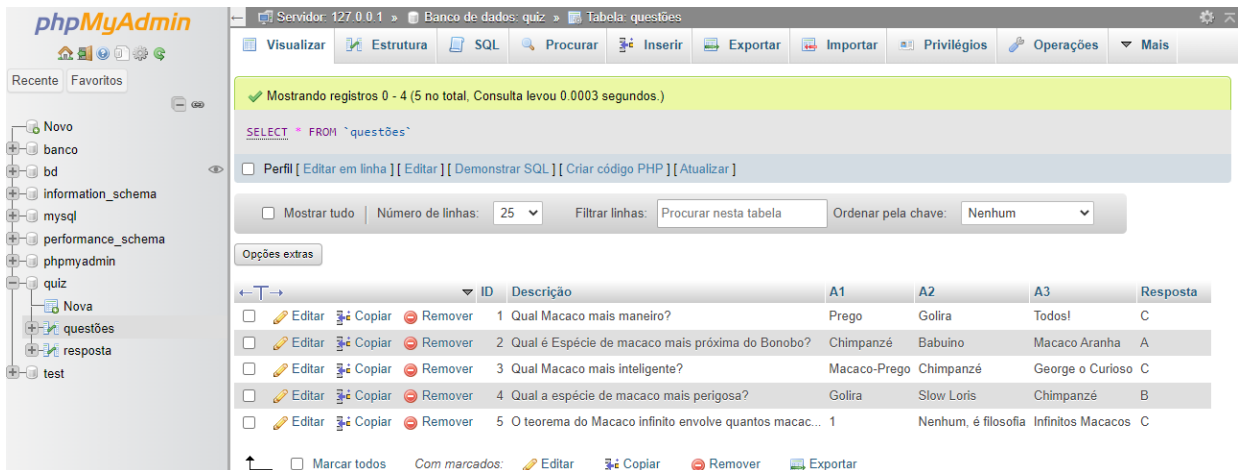
```
1 1
2 using System;
3 using System.Collections.Generic;
4 using System.Linq;
5 using System.Text;
6 using System.Threading.Tasks;
7
8 namespace QuizRetorneAo_Macaco
9 {
10     21 referências
11     class VariaveisGlobais
12     {
13         public static int Acertos = 0, Erros = 0;
14     }
15 }
```



## 2.2 Funcionamento para o Banco de Dados

Utilizando o XAMPP no meu banco de dados criei duas tabelas:

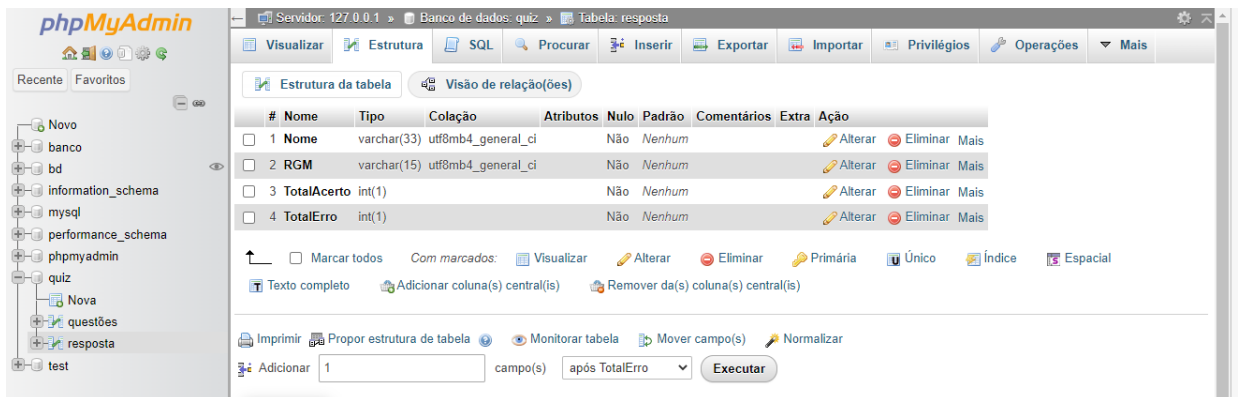
### Questões



The screenshot shows the phpMyAdmin interface for the 'quiz' database. The 'questões' table is selected, and the 'Visualizar' (View) tab is active. The table structure is shown as a SQL query: `SELECT * FROM `questões``. The table has 5 records. The columns are: ID, Descrição, A1, A2, A3, and Resposta.

ID	Descrição	A1	A2	A3	Resposta
1	Qual Macaco mais maneiro?	Prego	Golira	Todos!	C
2	Qual é Espécie de macaco mais próxima do Bonobo?	Chimpanzé	Babuino	Macaco Aranha	A
3	Qual Macaco mais inteligente?	Macaco-Prego	Chimpanzé	George o Curioso	C
4	Qual a espécie de macaco mais perigosa?	Golira	Slow Loris	Chimpanzé	B
5	O teorema do Macaco infinito envolve quantos macac...	1	Nenhum, é filosofia	Infinitos Macacos	C

### Resposta



The screenshot shows the phpMyAdmin interface for the 'quiz' database. The 'resposta' table is selected, and the 'Estrutura da tabela' (Table Structure) tab is active. The table structure is shown as a table with 4 columns: Nome, RGM, TotalAcerto, and TotalErro.

#	Nome	Tipo	Colação	Atributos	Nulo	Padrão	Comentários	Extra	Ação
1	Nome	varchar(33)	utf8mb4_general_ci		Não	Nenhum			Alterar Eliminar Mais
2	RGM	varchar(15)	utf8mb4_general_ci		Não	Nenhum			Alterar Eliminar Mais
3	TotalAcerto	int(1)			Não	Nenhum			Alterar Eliminar Mais
4	TotalErro	int(1)			Não	Nenhum			Alterar Eliminar Mais

### 3 Códigos:

Segue demonstração do Código de um forms, a consulta e o final onde guardamos o valor e enviamos para o DB

Primeiro, a tela do Menu:

```

13 public partial class Questionario : Form
14 {
15     2 referências
16     public Questionario()
17     {
18         InitializeComponent();
19     }
20     1 referência
21     private void Questionario_Load(object sender, EventArgs e)
22     {
23         //Zerar as variaveis no inicio
24         VariaveisGlobais.Acertos = 0;
25         VariaveisGlobais.Erros = 0;
26     }
27     0 referências
28     private void label2_Click(object sender, EventArgs e)
29     {
30     }
31
32     1 referência
33     private void button1_Click(object sender, EventArgs e)
34     {
35         //Aceitar iniciar o aplicativo
36
37         Questionario_Q1_Q1 = new Questionario_Q1();
38         //Fechar este e abrir o quiz
39         this.Hide();
40         Q1.ShowDialog();
41     }
42
43     1 referência
44     private void consultarOsResultadosToolStripMenuItem_Click(object sender, EventArgs e)
45     {
46         //Consultar o resultado pelo MenuStrip
47         Resultados rs = new Resultados();
48         rs.ShowDialog();

```

```

44     1 referência
45     private void consultarOsResultadosToolStripMenuItem_Click(object sender, EventArgs e)
46     {
47         //Consultar o resultado pelo MenuStrip
48         Resultados rs = new Resultados();
49         rs.ShowDialog();
50     }
51     1 referência
52     private void button2_Click(object sender, EventArgs e)
53     {
54         //Fechar o app
55         Application.Exit();
56     }
57 }
58

```

Com isto agora vamos ver um forms de exemplo:

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace QuizRetorneAo_Macaco
12 {
13     public partial class Questionario_Q1_ : Form
14     {
15
16
17
18         public Questionario_Q1_()
19         {
20             InitializeComponent();
21         }
22
23         private void Questionario_Q1_Load(object sender, EventArgs e)
24         {
25             //Instancio o Questoes e o QuestDAO
26             QuestoesDAO questdao = new QuestoesDAO();
27             Questoes quest = new Questoes();
28             //Defino meu ID como 1
29             quest.SetID(1);
30
31             //Puxo meu metodo consultar ID
32             questdao.consultarId(quest);
33             //Defino qual opção seria o que de acordo com DB
34             label1.Text = quest.GetID() + ". " + quest.GetDescricao();
35             radioButton1.Text = quest.GetA1();
36             radioButton2.Text = quest.GetA2();
37             radioButton3.Text = quest.GetA3();
38         }
39     }

```

```

41     private void radioButton3_CheckedChanged(object sender, EventArgs e)
42     {
43     }
44
45
46     private void button1_Click(object sender, EventArgs e)
47     {
48         //Instanciando
49         Respostas resp = new Respostas();
50         Questionario_Q2_ Q2 = new Questionario_Q2_();
51
52         //Logica de respostas
53
54         if (radioButton1.Checked == true) //Resposta Falsa
55         {
56             MessageBox.Show("Errou!");
57             VariaveisGlobais.Erros += 1; // Somo a variavel com mais um Erro
58             this.Hide(); //Escondo este app
59             Q2.ShowDialog(); //Passo para o prox forms
60         }
61         else
62         {
63             if (radioButton2.Checked == true) {
64
65                 //Soma da variavel global erros
66                 VariaveisGlobais.Erros += 1;
67                 MessageBox.Show("Errou!");
68                 this.Hide();
69                 Q2.ShowDialog();
70             }
71             else
72             {
73                 if (radioButton3.Checked == true)
74                 {
75                     //Soma da variavel global acertos
76                     VariaveisGlobais.Acertos += 1;
77                     MessageBox.Show("Acertou!");
78                     this.Hide();
79                     Q2.ShowDialog();

```

```

else
{
    if (radioButton3.Checked == true)
    {
        //Soma da variavel global acertos
        VariaveisGlobais.Acertos += 1;
        MessageBox.Show("Acertou!");
        this.Hide();
        Q2.ShowDialog();
    }
    else
    {
        MessageBox.Show("Escolha uma alternativa!"); //Caso não selecionem uma resposta
    }
}
}
}
}

```

Agora a consulta de valores:

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace QuizRetorneAo_Macaco
12 {
13     5 referências
14     public partial class Resultados : Form
15     {
16         1 referência
17         public Resultados()
18         {
19             InitializeComponent();
20         }
21
22         1 referência
23         private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
24         {
25         }
26
27         1 referência
28         private void Resultados_Load(object sender, EventArgs e)
29         {
30             dataGridView1.Rows.Clear(); // Limpo meu Grid antes de levar
31             QuestoesDAO qd = new QuestoesDAO(); //Instancia
32             dataGridView1.DataSource = qd.consultarRespostas(); //Definimos que o nosso DataGrid vai utilizar o Metodo de consulta
33         }
34     }
35 }

```

E finalmente a tela final de resultados:

```

1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Linq;
7  using System.Text;
8  using System.Threading.Tasks;
9  using System.Windows.Forms;
10
11 namespace QuizRetorneAo_Macaco
12 {
13     5 referências
14     public partial class Questionario_Resultado_ : Form
15     {
16         public int test;
17         1 referência
18         public Questionario_Resultado_()
19         {
20             InitializeComponent();
21         }
22
23         1 referência
24         private void Questionario_Resultado__Load(object sender, EventArgs e)
25         {
26             Respostas resp = new Respostas(); // Instancia
27
28             //Nós dois estou exibindo o valor da variavel para o usuario nas labels ocultas
29             label5.Text = VariaveisGlobais.Acertos.ToString();
30             label6.Text = VariaveisGlobais.Erros.ToString();
31         }
32
33         1 referência
34         private void button1_Click(object sender, EventArgs e) // Ao clicar para exibir "Meus resultados"
35         {
36             if(textBox1.Text == string.Empty) { //Se estiver vazio o nome ele pede para preencher
37                 MessageBox.Show("Preencha o seu Nome");
38             }
39         }
40     }
41 }

```

```

1 referência
private void button1_Click(object sender, EventArgs e) // Ao clicar para exibir "Meus resultados"
{
    if(textBox1.Text == string.Empty) { //Se estiver vazio o nome ele pede para preencher
        MessageBox.Show("Preencha o seu Nome");
    }
    else
    {
        if(textBox2.Text == string.Empty) //Se estiver vazio o RGM ele pede para preencher
        {
            MessageBox.Show("Preencha o seu RGM");
        }
        else //Se estiver tudo certo
        {
            //Torno as labels exibindo resultado visiveis para meu usuario
            label5.Visible = true;
            label6.Visible = true;
            //
            //Instanciando as classes
            QuestoesDAO qd = new QuestoesDAO();
            Respostas resp = new Respostas();
            //
            //Setando todos atributos do "Resposta"
            resp.SetNome(textBox1.Text);
            resp.SetRGM(textBox2.Text);
            resp.SetTotalAcerto(VariaveisGlobais.Acertos);
            resp.SetTotalErro(VariaveisGlobais.Erros);
            //
            //Enviando para o DB
            qd.Cadastrar(resp);
            //
            //Tornando o botão de retornar ao menu visivel
            button2.Visible = true;
        }
    }
}

```

```

1 referência
private void button2_Click(object sender, EventArgs e) // Botao voltar menu
{
    //Voltar ao menu(Fechar este forms e abrir o menu)
    Questionario Principal = new Questionario();
    this.Hide();
    Principal.ShowDialog();
}

```

Além disto também foi deixado uma Assinatura no Program.cs

```

1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Threading.Tasks;
5 using System.Windows.Forms;
6
7 namespace QuizRetorneAo_Macaco
8 {
9
10     // NICOLLAS SOUZA
11     // SISTEMAS DE INFORMAÇÃO - NOITE
12     // MADRUGADA - 26/05/2023
13     0 referências
14     static class Program
15     {
16         /// <summary>
17         /// Ponto de entrada principal para o aplicativo.
18         /// </summary>
19         [STAThread]
20         0 referências
21         static void Main()
22         {
23             Application.EnableVisualStyles();
24             Application.SetCompatibleTextRenderingDefault(false);
25             Application.Run(new Questionario());
26         }
27     }
28 }

```

#### 4 Funcionamento do aplicativo:

Com isto vamos ver como o aplicativo funciona:

Ao Abrir ele vai nós perguntar se queremos iniciar:

- Sim(Inicia o Quiz)
- Não(Encerra o app)


Em cima contem ainda um Menu strip onde podemos consultar resultados já feitos:



Clicando no strip:



É carregado este grid que pode ser fechado para ainda podermos jogar o Quiz

 Resultados

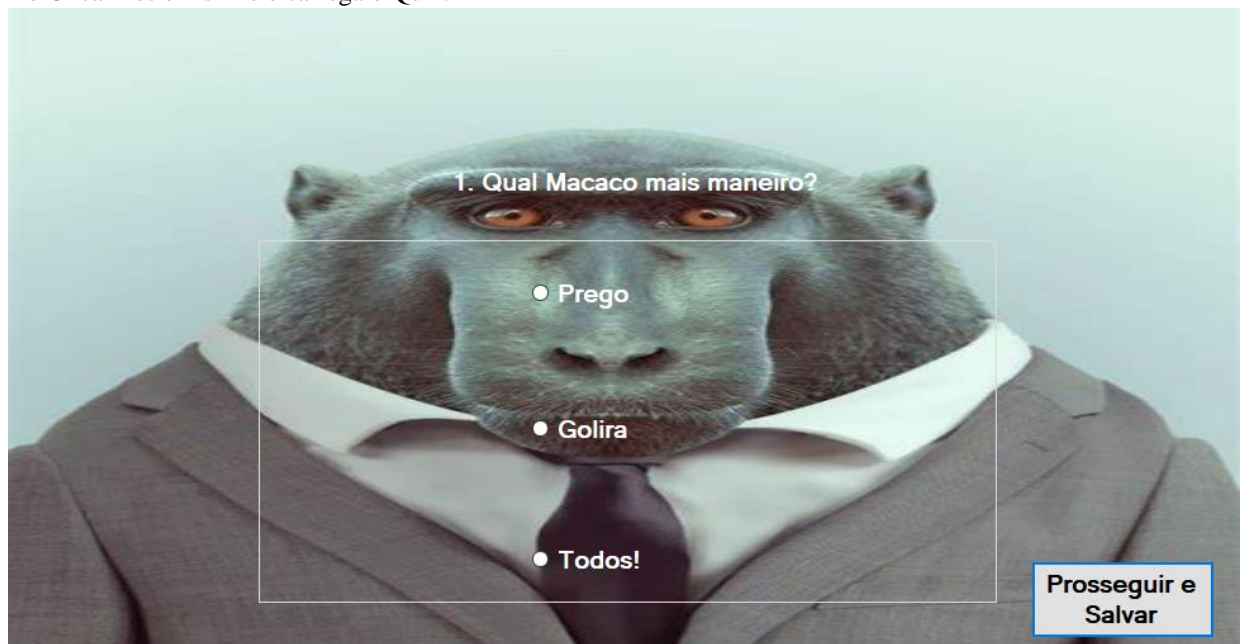
—

□

×

	Nome	RGM	TotalAcerto	TotalErro
▶	Teste	1121	0	0
	Teste2	2335	3	2
*				

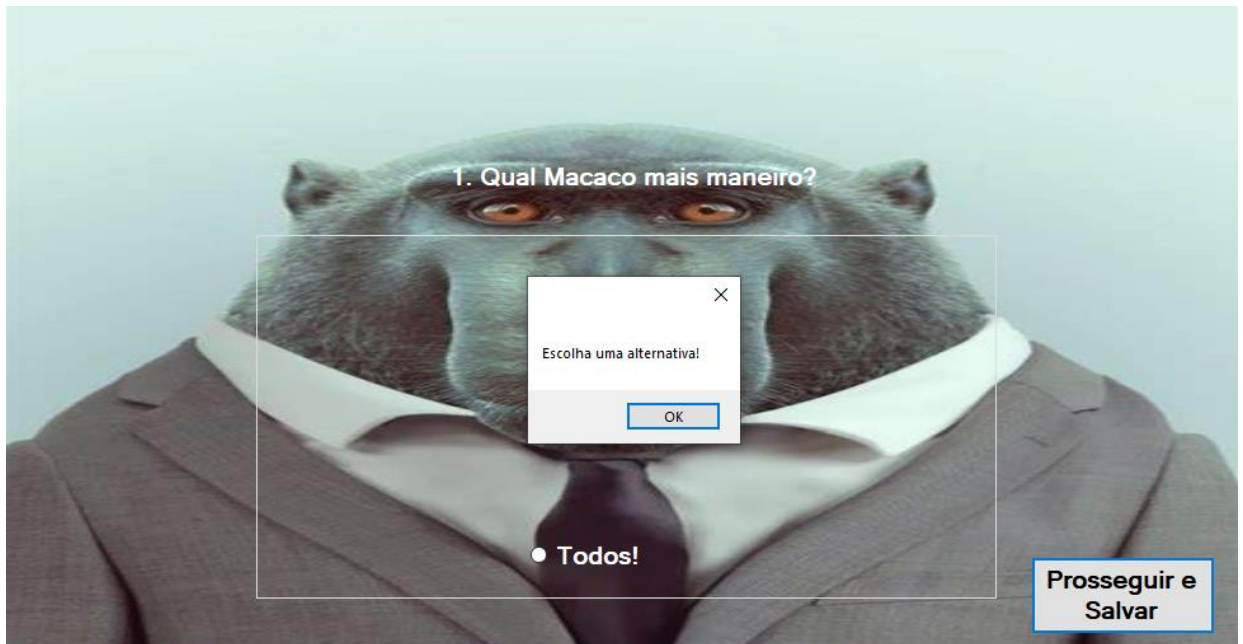
Ao Clicarmos em sim ele carrega o Quiz:



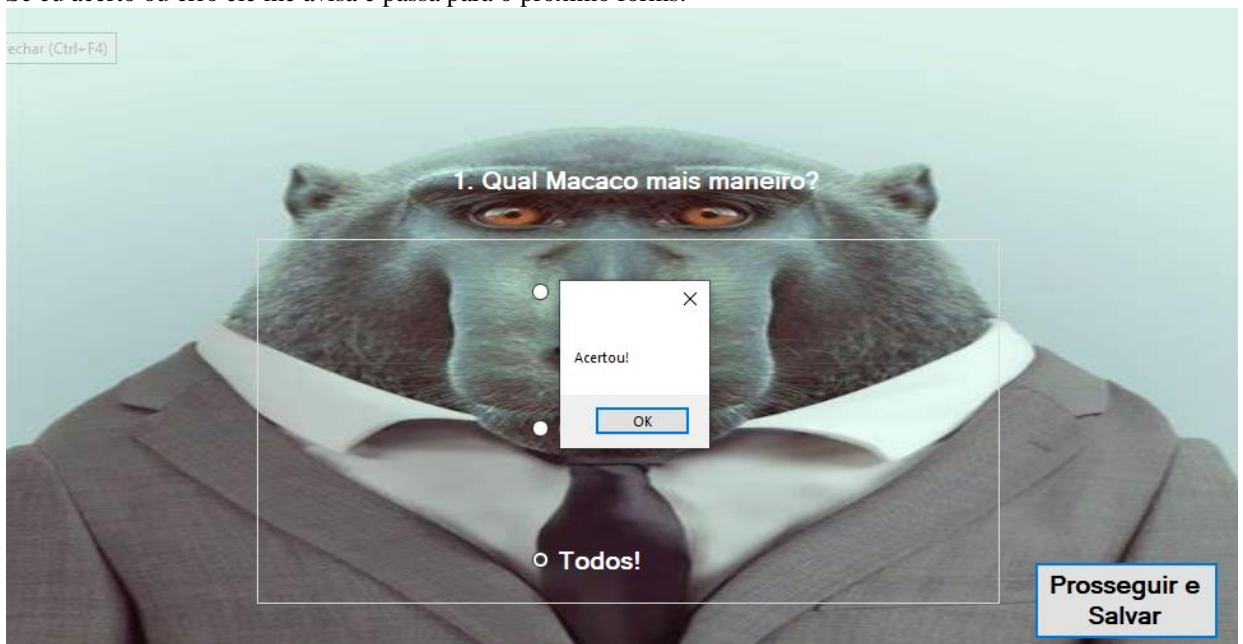
Detalhe: Não possui opção de fechar, assim que iniciar o quiz será obrigatório termina-lo



Se eu não seleccionar nenhuma alternativa:

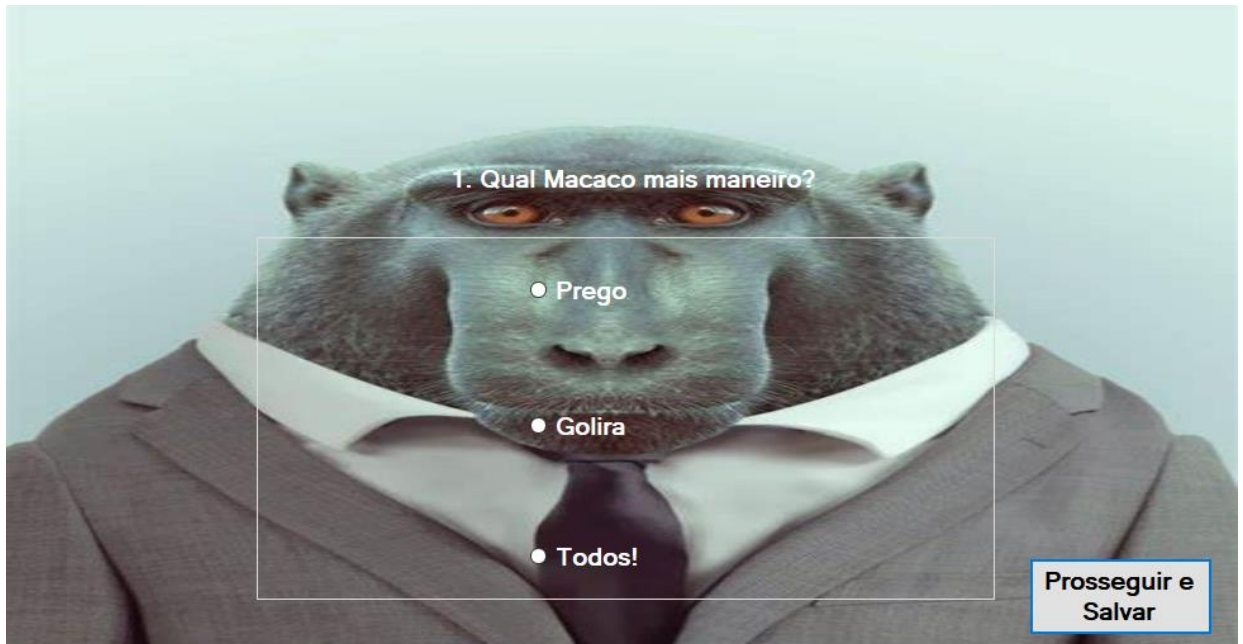


Se eu acerto ou erro ele me avisa e passa para o próximo forms:



Com o funcionamento do Form em mente segue todos em ordem

1.



2.

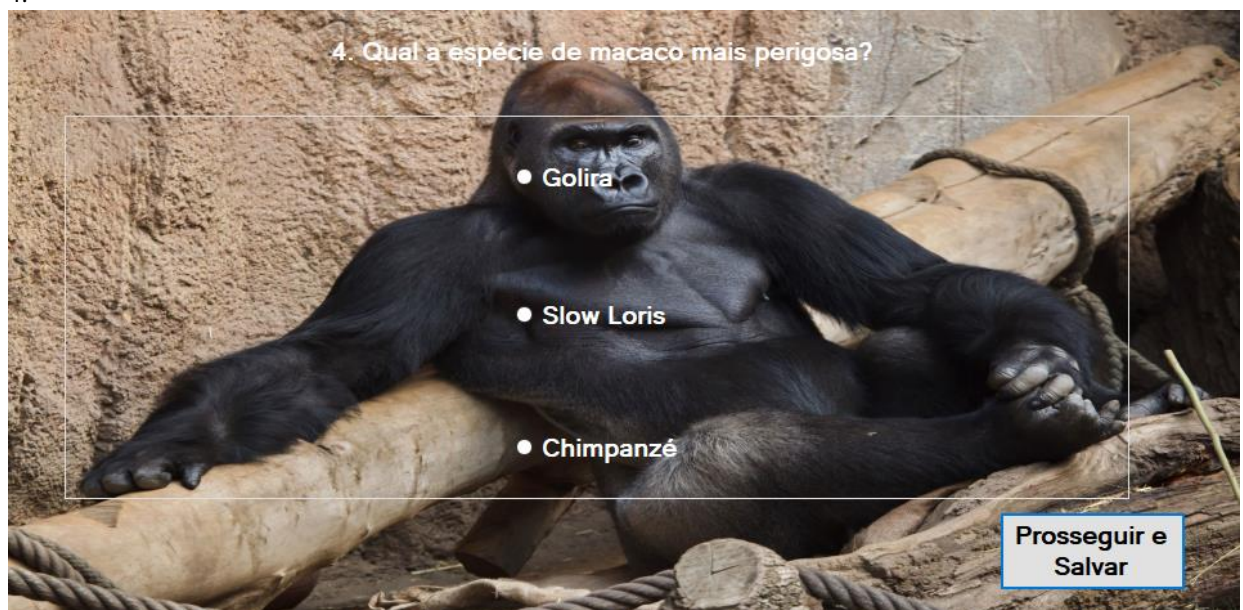




3.



4.



5.





Ao Respondermos esta ultima questão ele vai abrir a tela de resultado final, para ele te exibir seu resultado, seria necessário colocar o Nome e o RGM:

Coloque seu nome:

Coloque seu Rgm:

Preencha o campo acima para aparecer seus resultados

**Mostrar meu Resultado**

Acertos:

Erros:



Caso não preencha o nome nem o RGM ele te avisa

Questionario\_Resultado\_ — □ ×

Coloque seu nome:

Coloque seu Rgm:

Preencha o campo acima para aparecer seus resultados

**Mostrar meu Resultado**

Preencha o seu Nome

OK

Acertos:

Erros:



Assim que preenchido e consultar ele vai salvar no Banco de dados, aparecer a quantia de acertos/erros, e aparecer o botão para voltar ao menu

Questionario\_Resultado\_ — □ ×

Coloque seu nome:

Coloque seu Rgm:

Preencha o campo acima para aparecer seus resultados

**Mostrar meu Resultado**

Acertos: 3

Erros: 2

**Voltar ao Menu**



## **5 Conclusão:**

Ao Final deste aplicativo consegui praticar diversos conceitos obtidos em aula e fora dela como o Próprio DAO e as Variáveis globais, admito que tive bastante dificuldade durante o processo porém com o tempo acabei conseguindo realizar após insistir.