



UNIVERSIDADE DO VALE DO ITAJAÍ
CURSO SUPERIOR EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS -
FLORIANÓPOLIS

MIGUEL VERISSIMO JOSÉ MARTINS JÚNIOR

RELATÓRIO ANALÍTICO
PROJETO INTEGRADOR

FLORIANÓPOLIS

2020.1

MIGUEL VERISSIMO JOSÉ MARTINS JÚNIOR

RELATÓRIO ANALÍTICO

PROJETO INTEGRADOR

Relatório analítico referente ao Projeto Integrador para obtenção de avaliação para a Disciplina de Hands on Work III – Universidade do Vale do Itajaí – Curso superior em Análise e desenvolvimento de sistemas.

Professor: Alex Luciano Roesler Rese

FLORIANÓPOLIS

2020.2

Relatório Analítico – Projeto Integrador

ABSTRACT

The objective of this integrative project is to promote the content learned in the subjects, Programming and Data Persistence and Requirements Engineering. Thus building a functional system in C # with the standards that will be described by the Requirements Engineering during the 2020.2 semester.

RESUMO

O objetivo deste projeto integrador é fomentar os conteúdos aprendidos nas matérias, Programming and Data Persistence e Engenharia de Requisitos. Assim construindo um sistema funcional em C# com os padrões que serão descritos pela Engenharia de Requisitos durante o semestre 2020.2.

Palavras-chave: Projeto integrador. Sistema C#.

1 INTRODUÇÃO

O projeto integrador, partiu da idéia de manter os registros de nossos clientes com suas compras na linha de perfumaria da NATURA SA, onde somos consultores a minha esposa , Karine Rodrigues de Souza, e a minha pessoa.

O qual será inicialmente executado em C# e depois será transformado para um sistema multiplataforma.

2 DESENVOLVIMENTO

Durante o desenvolvimento do Front-end foram aplicadas as técnicas de UX Design para aumentar a interação entre o artefato e os usuários.

Foram utilizados para o desenvolvimento, as linguagens de programação C# no Visual Studio) para a estruturação do artefato e utilizado o mysql .

Desta forma, o Artefato podem ser acessados e baixados através do endereço no github <https://github.com/migueljr01/Natura.git> e a explicação do artefato através do Link <http://youtu.be/Yt38FRggCdI?hd=1>

Foi utilizado como base os trabalhos executados na aula como layout, conforme figura1.

Código	Nome	Endereço	Bairro	Cidade	CEP	E-mail	CPF	Identidade	Max Parcelado
1	Miguel Verissimo Jose Martins Junior	Servidão Leonel Domingos da Silva, 93	Inglese do Rio V...	Florianópolis	88058-730	test@test.com	9999999999999999	9999999999999999	1000
2	Karine Rodrigues de Souza	Servidão Leonel Domingos da Silva, 93	Inglese do Rio V...	Florianópolis	88058730	test@test.com	99999999	9999999999999999	9000

Figura 1

2.1 Especificações do Programa

Identificador :	<input type="text" value="RN1"/>	Descrição :	<input type="text" value="Cadastrar os Clientes para que possamos termos o histórico das compras e sugerir novos produtos."/>
Modulo :	<input type="text" value="Modelo de negócio"/>		
Data criação :	<input type="text" value="20/09/20"/>	Autor:	<input type="text" value="Miguel Martins Jr"/>
Data de alteração :	<input type="text"/>	Autor:	<input type="text"/>
Versão	<input type="text" value="1"/>	Dependencia :	<input type="text"/>

Identificador :	<input type="text" value="RF1"/>	Descrição :	<input type="text" value="Cadastrar Clientes com seus dados Pessoais (Nome Completo, Identidade , CPF e endereço) com telefone e email. Além de informar um Limite Maximo pra compras Parceladas"/>
Modulo :	<input type="text" value="Cadastro de Clientes"/>		
Data criação :	<input type="text" value="20/09/20"/>	Autor:	<input type="text" value="Miguel Martins Jr"/>
Data de alteração :	<input type="text"/>	Autor:	<input type="text"/>
Versão	<input type="text" value="1"/>	Dependencia :	<input type="text" value="RN1"/>

Identificador :	<input type="text" value="RN1"/>	Descrição :	<input type="text" value="Cadastro de Vendas a vista e Vendas parceladas dentro do Limite maximo pré estabelecido durante cadastro do cliente"/>
Modulo :	<input type="text" value="Cadastro de Vendas"/>		
Data criação :	<input type="text" value="20/09/20"/>	Autor:	<input type="text" value="Miguel Martins Jr"/>
Data de alteração :	<input type="text"/>	Autor:	<input type="text"/>
Versão	<input type="text" value="1"/>	Dependencia :	<input type="text" value="RF1"/>

Identificador :	<input type="text" value="RN2"/>		Descrição : Estar acessível, nas plataformas PC, Celular/Tablet e WEB.
Modulo :	<input type="text" value="Acessibilidade"/>		
Data criação :	<input type="text" value="20/09/20"/>	Autor: <input type="text" value="Miguel Martins Jr"/>	
Data de alteração :	<input type="text"/>	Autor: <input type="text"/>	
Versão	<input type="text" value="1"/>	Dependencia : <input type="text" value="RF1"/>	

2.2 Código Fonte

Segue o código fonte do cadastro de Clientes.

Inicio

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using MySql.Data.MySqlClient;

namespace Controle_Clientes
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        private void Form1_Load(object sender, EventArgs e)
        {
            // TODO: esta linha de código carrega dados na tabela
            'naturaDataSet.clientes'. Você pode movê-la ou removê-la conforme necessário.
            this.clientesTableAdapter.Fill(this.naturaDataSet.clientes);
            atualizarGrid();
        }
        private void limparCampos()
        {
            textBoxCodigo.Clear();
            textBoxNome.Clear();
            textBoxEndereco.Clear();
            textBoxBairro.Clear();
        }
    }
}
```

```

        textBoxCidade.Clear();
        textBoxCEP.Clear();
        textBoxEmail.Clear();
        textBoxCPF.Clear();
        textBoxIdentidade.Clear();
        textBoxParcelado.Clear();
    }

    private void atualizarGrid()
    {
        //Crio a estrutura da conexão com o banco e passa os parametros
        MySqlConnectionStringBuilder conexaoBD = new
        MySqlConnectionStringBuilder();
        conexaoBD.Server = "localhost";
        conexaoBD.Database = "natura";
        conexaoBD.UserID = "root";
        conexaoBD.Password = "";
        //Realizo a conexão com o banco
        MySqlConnection realizaConexaoBD = new
        MySqlConnection(conexaoBD.ToString());
        try
        {
            realizaConexaoBD.Open();

            MySqlCommand comandoMySQL = realizaConexaoBD.CreateCommand();
            comandoMySQL.CommandText = "SELECT * FROM `clientes` WHERE
ativoCliente=1"; //Traz todo mundo da tabela autor
            MySqlDataReader reader = comandoMySQL.ExecuteReader();

            dataGridView.Rows.Clear(); //FAZ LIMPAR A TABELA

            while (reader.Read())
            {
                DataGridViewRow row =
                (DataGridViewRow)dataGridView.Rows[0].Clone(); //FAZ UM CAST E CLONA A LINHA DA TABELA
                row.Cells[0].Value = reader.GetInt32(0); //ID
                row.Cells[1].Value = reader.GetString(1); //NOME
                row.Cells[2].Value = reader.GetString(2); //ENDEREÇO
                row.Cells[3].Value = reader.GetString(3); //BAIRRO
                row.Cells[4].Value = reader.GetString(4); //CIDADE
                row.Cells[5].Value = reader.GetString(5); //CEP
                row.Cells[6].Value = reader.GetString(6); //EMAIL
                row.Cells[7].Value = reader.GetString(7); //CPF
                row.Cells[8].Value = reader.GetString(8); //IDENTIDADE
                row.Cells[9].Value = reader.GetString(9); //VALOR
                dataGridView.Rows.Add(row); //ADICIONO A LINHA NA TABELA
            }

            realizaConexaoBD.Close();
            //MessageBox.Show("Removido com sucesso"); ;
        }
        catch (Exception ex)
        {
            MessageBox.Show("Can not open connection ! ");
            Console.WriteLine(ex.Message);
        }
    }

    private void dataGridView_CellContentClick_1(object sender,
    DataGridViewCellEventArgs e)

```



```

    {
        if (dataGridView.Rows[e.RowIndex].Cells[e.ColumnIndex].Value != null)
        {
            limparCampos();
            dataGridView.CurrentRow.Selected = true;
            //preenche os textbox com as células da linha selecionada
            textBoxCodigo.Text =
dataGridView.Rows[e.RowIndex].Cells["ColumnId"].FormattedValue.ToString();
            textBoxNome.Text =
dataGridView.Rows[e.RowIndex].Cells["ColumnNome"].FormattedValue.ToString();
            textBoxEndereco.Text =
dataGridView.Rows[e.RowIndex].Cells["ColumnEndereco"].FormattedValue.ToString();
            textBoxBairro.Text =
dataGridView.Rows[e.RowIndex].Cells["ColumnBairro"].FormattedValue.ToString();
            textBoxCidade.Text =
dataGridView.Rows[e.RowIndex].Cells["ColumnCidade"].FormattedValue.ToString();
            textBoxCEP.Text =
dataGridView.Rows[e.RowIndex].Cells["ColumnCEP"].FormattedValue.ToString();
            textBoxCPF.Text =
dataGridView.Rows[e.RowIndex].Cells["ColumnCPF"].FormattedValue.ToString();
            textBoxIdentidade.Text =
dataGridView.Rows[e.RowIndex].Cells["ColumnIdentidade"].FormattedValue.ToString();
            textBoxEmail.Text =
dataGridView.Rows[e.RowIndex].Cells["ColumnEmail"].FormattedValue.ToString();
            textBoxParcelado.Text =
dataGridView.Rows[e.RowIndex].Cells["ColumnValor"].FormattedValue.ToString();
        }
    }

    private void buttonLimpar_Click(object sender, EventArgs e)
    {
        limparCampos();
    }

    private void buttonAdd_Click(object sender, EventArgs e)
    {
        //Crio a estrutura da conexão com o banco e passa os parametros
        MySqlConnectionStringBuilder conexaoBD = new
        MySqlConnectionStringBuilder();
        conexaoBD.Server = "localhost";
        conexaoBD.Database = "natura";
        conexaoBD.UserID = "root";
        conexaoBD.Password = "";
        //Realizo a conexão com o banco
        MySqlConnection realizaConexaoBD = new
        MySqlConnection(conexaoBD.ToString());
        try
        {
            realizaConexaoBD.Open(); //Abre a conexão com o banco
            //MessageBox.Show("Conexão Aberta!");

            MySqlCommand comandoMySQL = realizaConexaoBD.CreateCommand(); //Crio
            um comando SQL
            comandoMySQL.CommandText = "INSERT INTO `clientes` (`idCliente`,
`nomeCliente`, `enderecoCliente`, `bairroCliente`, `cidadeCliente`, `cepCliente`,
`emailCliente`, `identidadeCliente`, `cpfCliente`, `limiteCliente`, `ativoCliente`)
VALUES (NULL, '" + textBoxNome.Text + "', '" + textBoxEndereco.Text + "', '" +
textBoxBairro.Text + "', '" + textBoxCidade.Text + "', '" + textBoxCEP.Text + "', '" +
textBoxEmail.Text + "', '" + textBoxIdentidade.Text + "', '" + textBoxCPF.Text + "',
'" + textBoxParcelado.Text + "', '1')";
            comandoMySQL.ExecuteNonQuery();

```

```

        realizaConexaoBD.Close(); // Fecho a conexão com o banco
        MessageBox.Show("Inserido com sucesso"); //Exibo mensagem de aviso
        atualizarGrid();
        limparCampos();
    }
    catch (Exception ex)
    {
        //MessageBox.Show("Não foi possível abrir a conexão! ");
        Console.WriteLine(ex.Message);
    }
}

private void buttonApagar_Click(object sender, EventArgs e)
{
    //Crio a estrutura da conexão com o banco e passa os parametros
    MySqlConnectionStringBuilder conexaoBD = new
    MySqlConnectionStringBuilder();
    conexaoBD.Server = "localhost";
    conexaoBD.Database = "natura";
    conexaoBD.UserID = "root";
    conexaoBD.Password = "";
    //Realizo a conexão com o banco
    MySqlConnection realizaConexaoBD = new
    MySqlConnection(conexaoBD.ToString());
    try
    {
        realizaConexaoBD.Open(); //Abre a conexão com o banco
        //MessageBox.Show("Conexão Aberta!");

        MySqlCommand comandoMySQL = realizaConexaoBD.CreateCommand(); //Crio
        um comando SQL
        comandoMySQL.CommandText = " UPDATE `clientes` SET `ativoCliente` =
        '0' WHERE `idCliente` = " + textBoxCodigo.Text + ";";
        comandoMySQL.ExecuteNonQuery();

        realizaConexaoBD.Close(); // Fecho a conexão com o banco
        MessageBox.Show("Apagado com sucesso"); //Exibo mensagem de aviso
        atualizarGrid();
        limparCampos();
    }
    catch (Exception ex)
    {
        //MessageBox.Show("Não foi possível abrir a conexão! ");
        Console.WriteLine(ex.Message);
    }
}

private void buttonSalvar_Click(object sender, EventArgs e)
{
    //Crio a estrutura da conexão com o banco e passa os parametros
    MySqlConnectionStringBuilder conexaoBD = new
    MySqlConnectionStringBuilder();
    conexaoBD.Server = "localhost";
    conexaoBD.Database = "natura";
    conexaoBD.UserID = "root";
    conexaoBD.Password = "";
    //Realizo a conexão com o banco
    MySqlConnection realizaConexaoBD = new
    MySqlConnection(conexaoBD.ToString());
    try
    {
        realizaConexaoBD.Open(); //Abre a conexão com o banco

```

```

        //MessageBox.Show("Conexão Aberta!");

        MySqlCommand comandoMySQL = realizaConexaoBD.CreateCommand(); //Crio
um comando SQL
        comandoMySQL.CommandText = "UPDATE `clientes` SET `nomeCliente` = '" +
textBoxNome.Text + "', `enderecoCliente` = '" + textBoxEndereco.Text + "',
`bairroCliente` = '" + textBoxBairro.Text + "', `cidadeCliente` = '" +
textBoxCidade.Text + "', `cepCliente` = '" + textBoxCEP.Text + "', `emailCliente` = '" +
textBoxEmail.Text + "', `identidadeCliente` = '" + textBoxIdentidade.Text + "',
`limiteCliente` = '" + textBoxParcelado.Text + "' WHERE `idCliente` = " +
textBoxCodigo.Text + ";";
        comandoMySQL.ExecuteNonQuery();

        realizaConexaoBD.Close(); // Fecho a conexão com o banco
        MessageBox.Show("Salvo com sucesso"); //Exibo mensagem de aviso
        atualizarGrid();
        limparCampos();
    }
    catch (Exception ex)
    {
        //MessageBox.Show("Não foi possível abrir a conexão! ");
        Console.WriteLine(ex.Message);
    }
}
}
}
}

```

TERMINO

3 CONSIDERAÇÕES FINAIS

Este é um trabalho ainda em desenvolvimento que estaremos desenvolvendo este sistema durante as atividades do curso, assim estaremos desenvolvendo este artefato .

REFERÊNCIAS BIBLIOGRÁFICAS

Sem autor. Tutoriais diversos, 1999-2020. Disponível em: < <https://www.w3schools.com/> >.

Acesso em: 01 set 2020.

Alex Luciano Roesler Rese. Material das Aulas ministradas no curso de Programming and Data Persistence. Disponível no Blackboard. Acesso :04/08/2020 à 24/08/2020