

Universidade do Vale do Itajaí – UNIVALI
Escola do Mar, Ciência e Tecnologia
Curso de Análise e Desenvolvimento de Sistemas
Disciplina Hands on Work VI
Aluno: Paulo Luciano Silva dos Santos



Projeto de Hands on Work VI

Relatório
FINAL

**PROJETO E DESENVOLVIMENTO
DE SISTEMA COMPUTACIONAL
COM PERSISTÊNCIA DE DADOS**

Sumário

Sumário	2
Introdução.....	3
Fórum Temático	3
Modelagem do Projeto	3
Diagrama de Classe	3
MER – Modelo de Entidade de Relacionamento	4
Código e do Banco de Dados	5
Código MySql.....	5
Formulário do Visual Studio 2022.	6
Tela Principal	6
Tela Cliente	7
Tela Produto.....	7
Estrutura e código do Visual Studio.....	8
Formulário Principal	8
Formulário Cliente	9
Formulário Produto.....	11
Classe Banco.....	14
Classe Usuario.....	15
Vídeo de explicação do Sistema	17
Considerações.....	18
Referências Bibliográficas	19

Introdução

O sistema proposto, é para criar uma relação entre duas tabelas, e criar um sistema para fazer o CRUD no banco de dados.

Foi utilizado o SGBD MySQL, para o banco de dados e o sistema foi desenvolvido no Visual Studio 2022.

Fórum Temático

ATIVIDADE HANDS-ON

Projeto e Desenvolvimento de Sistema Computacional com Persistência de Dados

A) Defina uma entidade de dados e que possua uma segunda entidade vinculada. O tipo de vínculo deve configurar uma relação de composição ou agregação da orientação a objetos.

Exemplos: aluno x disciplinas; produto x categoria; matriz x filial; cliente x compra; etc.

* Cada entidade deve ter no mínimo 3 atributos – a ser definido pelo aluno.

B) **ETAPA 1:** Desenhe um diagrama de classes e também um diagrama de entidade relacionamento representando estas duas entidades, na perspectiva de Programação Orientada a Objetos e na perspectiva de tabelas, respectivamente.

C) **ETAPA 2:** Implemente interfaces web (telas) para realização de cadastro e listagem dos dados destas entidades interagindo com banco de dados. Por meio de telas o usuário deve ser capaz de:

- a. Incluir registros para cada uma das entidades; e
- b. Listar todos os registros de cada entidade.

A atividade pode ser feita de forma individual ou em dupla, e junto com a entrega do código funcional, deve ser entregue um vídeo de no máximo 4 minutos apresentando o software em funcionamento.

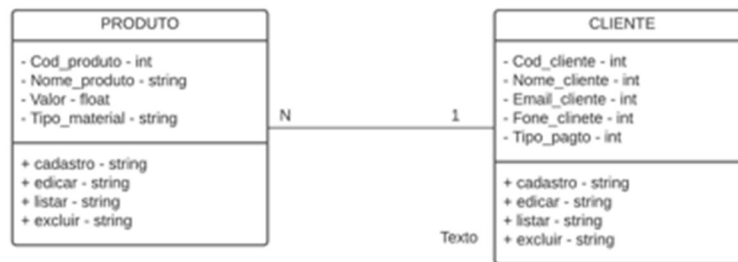
Modelagem do Projeto

Para fazer a modelagem utilizamos o Diagrama de Classe, MER para poder criar o banco de dados:

Diagrama de Classe

O diagrama de classe modela os objetos que compõem o sistema, exibindo os relacionamentos entre os objetos descrevendo estes objetos e os serviços que eles fornecem.

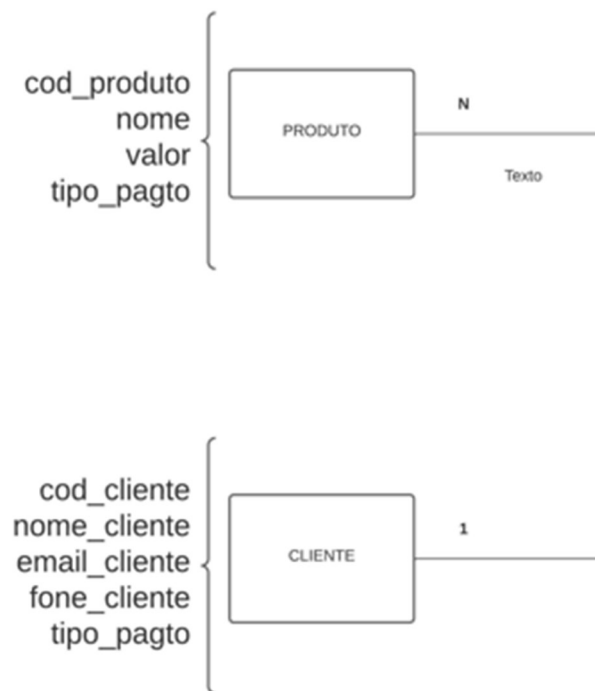
DIARAMA DE CLASSE



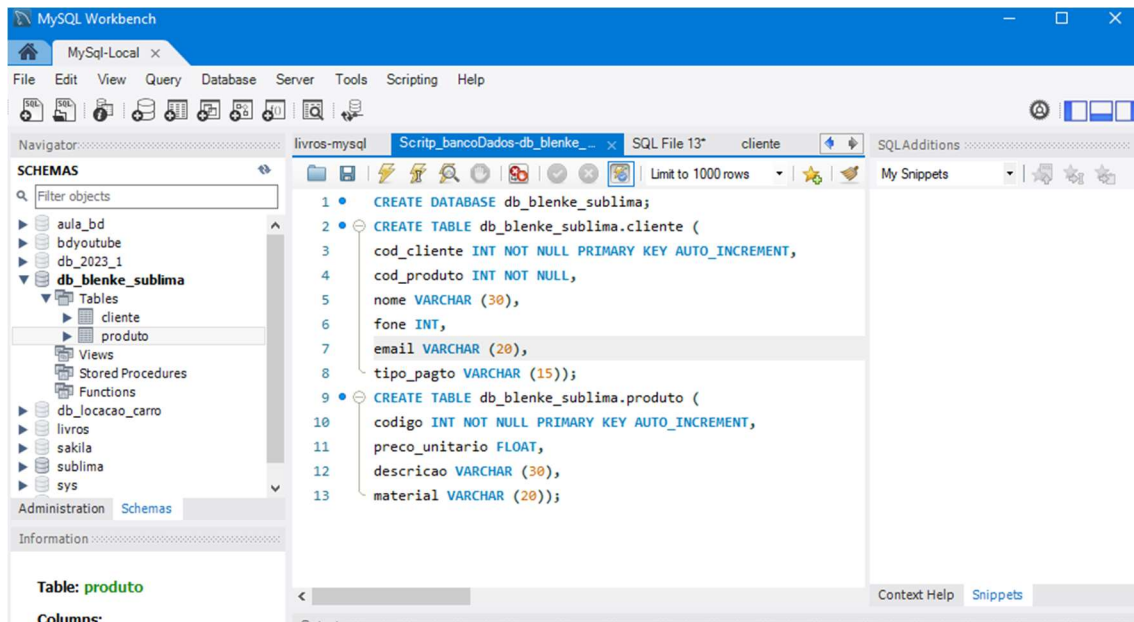
MER – Modelo de Entidade de Relacionamento

Esta representação do MER mostra a entidade e a existência de independências que o objetos possuem.

MER – Modelo de Entidade e Relacionamentos



Código e do Banco de Dados



Código MySQL

Comando utilizado para a criação do banco de dados:

```

CREATE DATABASE db_blenke_sublima;

CREATE TABLE db_blenke_sublima.cliente (
cod_cliente INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
cod_produto INT NOT NULL,
nome VARCHAR (30),
fone INT,
email VARCHAR (20),
tipo_pagto VARCHAR (15));

CREATE TABLE db_blenke_sublima.produto (
codigo INT NOT NULL PRIMARY KEY AUTO_INCREMENT,
preco_unitario FLOAT,
descricao VARCHAR (30),
material VARCHAR (20));

```

Formulário do Visual Studio 2022.

O desenvolvimento do sistema de computacional, foi utilizado o Visual Studio, o sistema tem um formulário principal como tela inicial do Sistema, e dois formulário secundários, um formulário com a parte Cliente e o outro com o Produto.

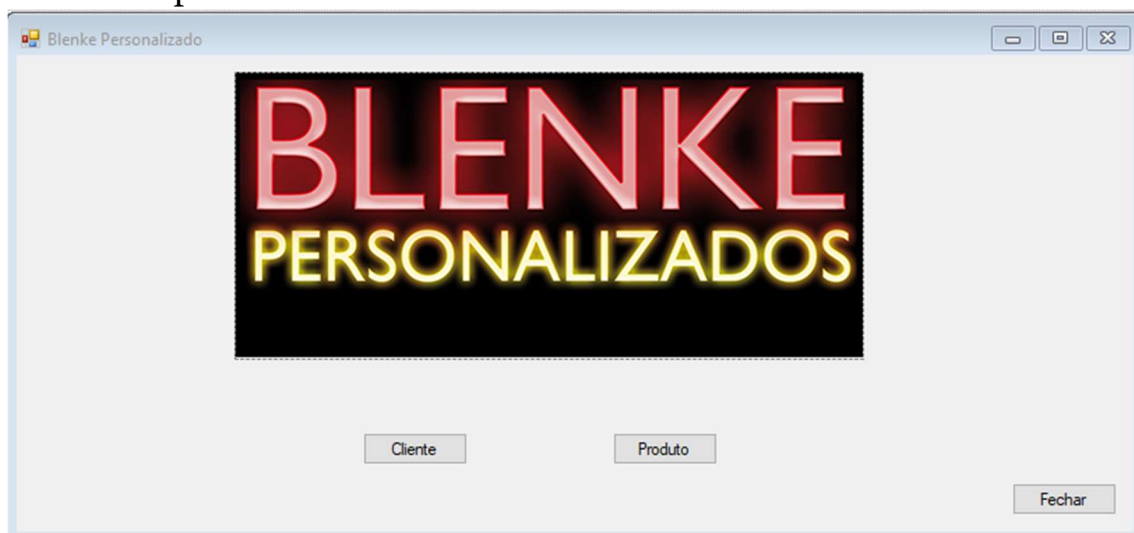
O sistema mostra os dados do banco de dados, o registro para ser selecionado, para Alteração ou Exclusão basta dar um duplo clique em cima do registro.

Para Inclusão, preencher os campos em branco e clicar no botão Cadastrar.

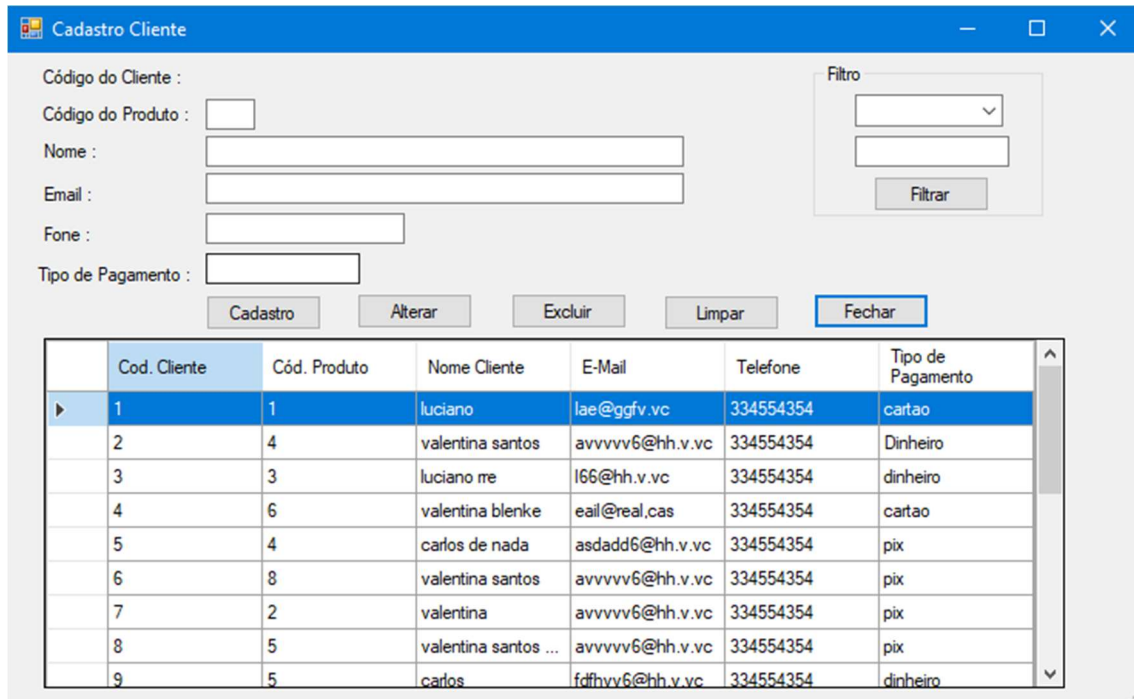
A opção Filtro, permite a possibilidade de fazer um filtro no banco e o mesmo seja mostrado no grid da tabela.

Abaixo segue as telas do formulários:

Tela Principal

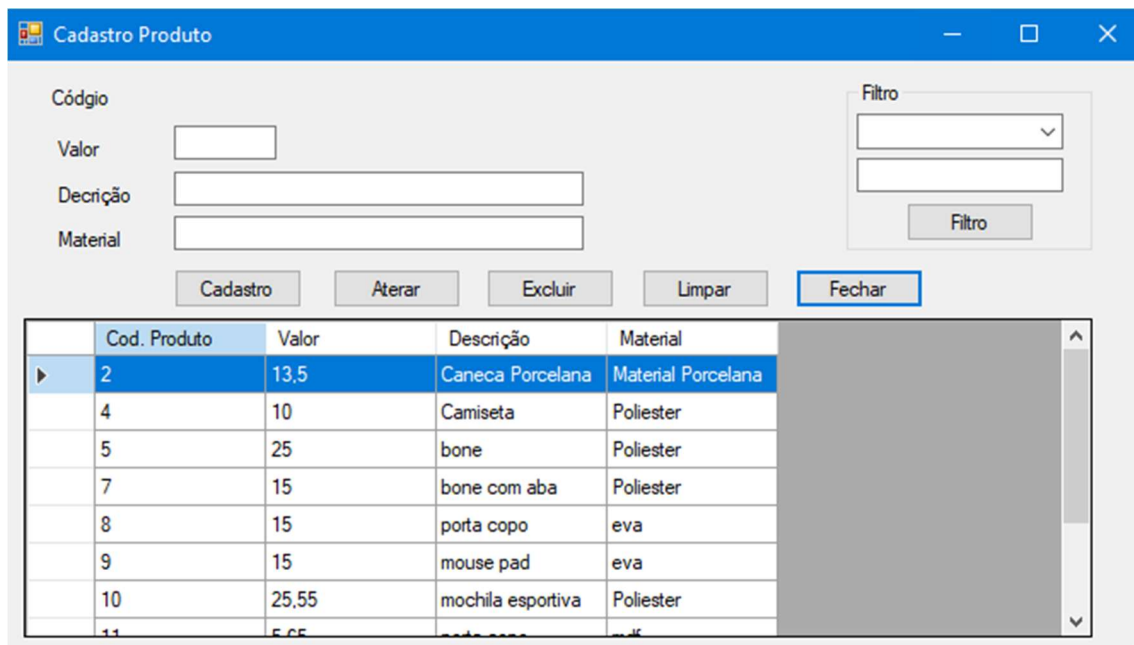


Tela Cliente



	Cod. Cliente	Cód. Produto	Nome Cliente	E-Mail	Telefone	Tipo de Pagamento
▶	1	1	luciano	lae@ggfv.vc	334554354	cartao
	2	4	valentina santos	avvvvv6@hh.v.vc	334554354	Dinheiro
	3	3	luciano me	l66@hh.v.vc	334554354	dinheiro
	4	6	valentina blenke	eail@real.cas	334554354	cartao
	5	4	carlos de nada	asdadd6@hh.v.vc	334554354	pix
	6	8	valentina santos	avvvvv6@hh.v.vc	334554354	pix
	7	2	valentina	avvvvv6@hh.v.vc	334554354	pix
	8	5	valentina santos ...	avvvvv6@hh.v.vc	334554354	pix
	9	5	carlos	fdhvv6@hh.v.vc	334554354	dinheiro

Tela Produto



	Cod. Produto	Valor	Descrição	Material
▶	2	13,5	Caneca Porcelana	Material Porcelana
	4	10	Camiseta	Poliester
	5	25	bone	Poliester
	7	15	bone com aba	Poliester
	8	15	porta copo	eva
	9	15	mouse pad	eva
	10	25,55	mochila esportiva	Poliester
	11	5,65	capa para...	poli...

Os formulários possuem a mesma estrutura, facilitando assim a manipulação dos dados, os campos podem ser preenchido, com as informações solicitadas e clicando no Botão Cadastro, será inserido um novo registro.

Dando um duplo click em cima de qualquer registro, os dados do registro aparece nos campos, dando a possibilidade de alterar ou excluir o registro, o Botão Limpar, serve para limpar os dados do formulário.

Na área Filtro pode-se filtra os dados do banco de dados, seleciona na caixa de seleção, e na linha de baixo coloca o filtro a ser executado.

Estrutura e código do Visual Studio

Foi criado duas (02) classes, uma para o Banco de Dados, onde estão o acesso e ao banco e a outra onde foi criado um usuário par a execução dos comandos sql.

Formulário Principal

```
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;

// foi instalado uma referência para poder utilizar os comandos do MySql e
conectar o banco de dados
// MySql.Data

namespace BnekePersonalizado
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        // botão para fechar e sair do formulário
        private void btnClose_Click(object sender, EventArgs e)
        {
            this.Close();
        }

        //botão para abrir o formulario Cliente
        private void btnCliente_Click(object sender, EventArgs e)
        {
            FormCliente abrirFormcli = new FormCliente();
            abrirFormcli.ShowDialog();
        }

        //botão para abrir o formulario Produto
        private void btnProduto_Click(object sender, EventArgs e)
        {
            FormProduto abrirFormProduto = new FormProduto();
            abrirFormProduto.ShowDialog();
        }
    }
}
```

Fim do código

Formulário Cliente

```
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 BlnেকেPersonalizado
{
    public partial class FormCliente : Form
    {
        public FormCliente()
        {
            InitializeComponent();

            // botão para fechar o formulário

            private void btnfechar_Click(object sender, EventArgs e)
            {
                this.Close();
            }

            // carrega o formulário e já carrega o metodo AtualizarDataGrid ( este
            método carrega informações
            // do datagrid com os dados do banco.

            private Usuario usuario;
            private void FormCliente_Load(object sender, EventArgs e)
            {
                this.usuario = new Usuario();
                AtualizarDataGrid();
            }

            //configura os títulos da tabela do banco de dados
            private void ConfigurarGridCliente()
            {
                dtGridCliente.Columns["cod_cliente"].HeaderText = "Cod. Cliente";
                dtGridCliente.Columns["cod_produto"].HeaderText = "Cód. Produto";
                dtGridCliente.Columns["nome"].HeaderText = "Nome Cliente";
                dtGridCliente.Columns["email"].HeaderText = "E-Mail";
                dtGridCliente.Columns["fone"].HeaderText = "Telefone";
                dtGridCliente.Columns["tipo_pagto"].HeaderText = "Tipo de Pagamento";
            }

            // método para preencher o datagrid com as informações do banco de dados.
            private void AtualizarDataGrid()
            {
                MySqlDataReader temp = this.usuario.ListarUsuario();
                DataTable dt = new DataTable();
                dt.Load(temp);
                dtGridCliente.DataSource = dt;
            }
        }
    }
}
```

```
        ConfigurarGridCliente();
    }

    // filtra dados do banco de dados
    private void btnFiltro_Click(object sender, EventArgs e)
    {
        string filtro = txtFiltroCli.Text;
        string campo = cboFiltroCli.Text;

        if (campo == "Código Produto")
        {
            campo = "cod_produto";
        }
        if (campo == "Tipo Pagamento")
        {
            campo = "tipo_pagto";
        }

        MySqlDataReader temp = this.usuario.ListarUsuario(campo, filtro);
        DataTable dt = new DataTable();
        dt.Load(temp);
        dtGridCliente.DataSource = dt;
    }

    private void label3_Click(object sender, EventArgs e)
    {
    }

    // metodo para pegar as informações do banco de dados e coloca em campo
    para serem alterado
    private void dtGridCliente_CellDoubleClick(object sender,
        DataGridViewCellEventArgs e)
    {
        DataGridViewRow linha = dtGridCliente.Rows[e.RowIndex];

        lblCodCli.Text = linha.Cells[0].Value.ToString();
        txtCodProd.Text = linha.Cells[1].Value.ToString();
        txtNomeCli.Text = linha.Cells[2].Value.ToString();
        txtEmail.Text = linha.Cells[3].Value.ToString();
        txtFone.Text = linha.Cells[4].Value.ToString();
        txtTipoPag.Text = linha.Cells[5].Value.ToString();
    }

    // limpa os campos do formulário
    private void LimparCampos()
    {
        txtCodProd.Text = "";
        lblCodCli.Text = "";
        txtNomeCli.Text = "";
        txtEmail.Text = "";
        txtFone.Text = "";
        txtTipoPag.Text = "";
        AtualizarDataGrid();
    }

    // botão para limpar o formulário
    private void btnLimpar_Click(object sender, EventArgs e)
    {
        LimparCampos();
    }
}
```

```
// cadastro de Cliente
private void btnCadastro_Click(object sender, EventArgs e)
{
    this.usuario.Cod_produto = txtCodProd.Text;
    this.usuario.Nome = txtNomeCli.Text;
    this.usuario.Email = txtEmail.Text;
    this.usuario.Fone = txtFone.Text;
    this.usuario.Tipo_pagto = txtTipoPag.Text;
    this.usuario.CadastrarUsuario();
    LimparCampos();
    AtualizarDataGrid();
}

// Alterar Cliente
private void btnAlterar_Click(object sender, EventArgs e)
{
    this.usuario.Cod_cliente = lblCodCli.Text;
    this.usuario.Cod_produto = txtCodProd.Text;
    this.usuario.Nome = txtNomeCli.Text;
    this.usuario.Email = txtEmail.Text;
    this.usuario.Fone = txtFone.Text;
    this.usuario.Tipo_pagto = txtTipoPag.Text;
    this.usuario.AlterarUsuario();
    LimparCampos();
}

// Excluir Cliente
private void btnExcluir_Click(object sender, EventArgs e)
{
    this.usuario.Cod_cliente = this.lblCodCli.Text;
    this.usuario.ExcluirUsusario();
    LimparCampos();
}
}
```

Fim do código

Formulário Produto

```
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 BlnেকেPersonalizado
{
    public partial class FormProduto : Form
    {
        public FormProduto()
        {
            InitializeComponent();
        }

        private Usuario usuario;
```

```
// botão para fechar o formulário
private void btnfechar_Click(object sender, EventArgs e)
{
    this.Close();
}

// carrega o formulário e já carrega o metodo AtualizarDataGrid ( este
método carrega informações
// do datagrid com os dados do banco.
private void FormProduto_Load(object sender, EventArgs e)
{
    this.usuario = new Usuario();
    AtualizarDataGrid();
}

//configura os títulos da tabela do banco de dados
private void ConfigurarGridProduto()
{
    dtGridProduto.Columns["codigo"].HeaderText = "Cod. Produto";
    dtGridProduto.Columns["preco_unitario"].HeaderText = "Valor";
    dtGridProduto.Columns["descricao"].HeaderText = "Descrição";
    dtGridProduto.Columns["material"].HeaderText = "Material";
}

// método para preencher o datagrid com as informações do banco de dados.
private void AtualizarDataGrid()
{
    MySqlDataReader temp = this.usuario.Listarproduto();
    DataTable dt = new DataTable();
    dt.Load(temp);
    dtGridProduto.DataSource = dt;
    ConfigurarGridProduto();
}

// filtra dados do banco de dados
private void btnFiltro_Click(object sender, EventArgs e)
{
    string filtro = txtfiltro.Text;
    string campo = cboFiltro.Text;

    if (campo == "Código")
    {
        campo = "codigo";
    }
    if (campo == "Descrição")
    {
        campo = "descricao";
    }

    MySqlDataReader temp = this.usuario.Listarproduto(campo, filtro);
    DataTable dt = new DataTable();
    dt.Load(temp);
    dtGridProduto.DataSource = dt;
}

// limpa os campos do formulário
private void LimparCampos()
{
    lblCodProd.Text = "";
    txtPrecoUni.Text = "";
    txtDescricao.Text = "";
}
```

```
        txtMaterial.Text = "";
        AtualizarDataGrid();
    }

    // botão para limpar o formulário
    private void btnLimpar_Click(object sender, EventArgs e)
    {
        LimparCampos();
    }

    // metodo para pegar as informações do banco de dados e coloca em campo
    para serem alterado
    private void dtGridProduto_CellMouseDoubleClick(object sender,
    DataGridViewCellMouseEventArgs e)
    {
        DataGridViewRow linha = dtGridProduto.Rows[e.RowIndex];
        lblCodProd.Text      = linha.Cells[0].Value.ToString();
        txtPrecoUni.Text     = linha.Cells[1].Value.ToString();
        txtDescricao.Text    = linha.Cells[2].Value.ToString();
        txtMaterial.Text     = linha.Cells[3].Value.ToString();
    }

    // cadastro de Produto
    private void btnCadastroPro_Click(object sender, EventArgs e)
    {
        this.usuario.Preco_unitario = txtPrecoUni.Text;
        this.usuario.Descricao = txtDescricao.Text;
        this.usuario.Material = txtMaterial.Text;
        this.usuario.CadastrarProduto();
        LimparCampos();
        AtualizarDataGrid();
    }

    // alterar produto
    private void btnAlterarProd_Click(object sender, EventArgs e)
    {
        this.usuario.Codigo = lblCodProd.Text;
        this.usuario.Preco_unitario = txtPrecoUni.Text;
        this.usuario.Descricao = txtDescricao.Text;
        this.usuario.Material = txtMaterial.Text;
        this.usuario.AlterarProduto();
        LimparCampos();
    }

    // excluir produto
    private void btnExcluirProd_Click(object sender, EventArgs e)
    {
        this.usuario.Codigo = this.lblCodProd.Text;
        this.usuario.ExcluirProduto();
        LimparCampos();
    }
}
}
```

Fim do código

Classe Banco

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using MySql.Data.MySqlClient;

namespace BlnekePersonalizado
{
    // parametros para conectar o banco de dados MySql
    class Banco
    {
        private string host = "localhost";
        private string nomeBanco = "db_blenke_sublima";
        private string usuario = "root";
        private string senha = "123456";
        private MySqlConnection con;
        private MySqlCommand command;

        public Banco()
        {
            Conectar();
        }

        // Método para conectar o banco de dados.
        public void Conectar()
        {
            String stringCon = "SERVER=" + this.host + ";DATABASE=" +
this.nomeBanco + ";UID=" + this.usuario + ";PASSWORD=" + this.senha + ";";
            this.con = new MySqlConnection(stringCon);
            this.command = this.con.CreateCommand();
            this.con.Open();
        }

        // Método para ser usado pelo DELETE, UPDATE
        public void NonQuery(string sql)
        {
            this.command.CommandText = sql;
            this.command.ExecuteNonQuery();
        }

        //Método para ser usado pelo INSERT, SELECT
        public MySqlDataReader Query(string sql)
        {
            this.command.CommandText = sql;
            return this.command.ExecuteReader();
        }
    }
}
```

Fim do código

Classe Usuario

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

namespace BlnেকেPersonalizado
{
    class Usuario
    {
        private string cod_produto;
        private string cod_cliente;
        private string nome;
        private string email;
        private string fone;
        private string tipo_pagto;
        private string codigo;
        private string preco_unitario;
        private string descricao;
        private string material;
        private Banco banco;

        public Usuario()
        {
            this.banco = new Banco();
        }

        // exclusão de Cliente
        public void ExcluirUsuario()
        {
            this.banco.NonQuery("DELETE FROM cliente WHERE cod_cliente = '" +
this.cod_cliente + "';");
        }

        // alterar Cliente
        public void AlterarUsuario()
        {
            this.banco.NonQuery("UPDATE cliente SET cod_produto = '" +
this.cod_produto + "', nome='" + this.nome + "',email='" + this.email +
"',fone='" + this.fone + "',tipo_pagto='" + this.tipo_pagto + "' WHERE
cod_cliente='" + this.cod_cliente + "';");
        }

        // cadastrar Cliente
        public void CadastrarUsuario()
        {
            this.banco.NonQuery("INSERT INTO cliente (cod_produto, nome, email,
fone, tipo_pagto) VALUE ('" + this.cod_produto + "', '" + this.nome + "', '" +
this.email + "', '" + this.fone + "', '" + this.tipo_pagto + "');");
        }

        //lista todos os clientes do banco
        public MySqlDataReader ListarUsuario()
        {
            return this.banco.Query("SELECT * FROM cliente");
        }
    }
}
```

```

    }

    // Método para filtrar clientes
    public MySqlDataReader ListarUsuario(string campo, string filtro)
    {
        if (filtro == "")
        {
            return ListarUsuario();
        }
        return this.banco.Query("SELECT * FROM cliente WHERE " + campo + " = '" + filtro + "'");
    }

    // mostrar todos os dados da tabela produto
    public MySqlDataReader Listarproduto()
    {
        return this.banco.Query("SELECT * FROM produto");
    }

    // filtro da tabela produto
    public MySqlDataReader Listarproduto(string campo, string filtro)
    {
        if (filtro == "")
        {
            return Listarproduto();
        }

        return this.banco.Query("SELECT * FROM produto WHERE " + campo + " = '" + filtro + "'");
    }

    // Método para cadastro de protudo
    public void CadastrarProduto()
    {
        this.banco.NonQuery("INSERT INTO produto (preco_unitario, descricao, material) VALUE ('" + this.preco_unitario + "','" + this.descricao + "','" + this.material + "')");
    }

    // Método para Alerar o protudo
    public void AlterarProduto()
    {
        this.banco.NonQuery("UPDATE produto SET preco_unitario='" + this.preco_unitario + "',material='" + this.material + "',descricao='" + this.descricao + "' WHERE codigo='" + this.codigo + "'");
    }

    // Método para Excluri o protudo
    public void ExcluirProduto()
    {
        this.banco.NonQuery("DELETE FROM produto WHERE codigo = '" + this.codigo + "'");
    }

    // encapsular campos que são utilizados
    public string Cod_produto { get => cod_produto; set => cod_produto = value; }
    public string Cod_cliente { get => cod_cliente; set => cod_cliente = value; }

```



```

        public string Nome { get => nome; set => nome = value; }
        public string Email { get => email; set => email = value; }
        public string Fone { get => fone; set => fone = value; }
        public string Tipo_pagto { get => tipo_pagto; set => tipo_pagto = value;
    }

    public string Codigo { get => codigo; set => codigo = value; }
    public string Preco_unitario { get => preco_unitario; set =>
preco_unitario = value; }
    public string Descricao { get => descricao; set => descricao = value; }
    public string Material { get => this.material; set => this.material =
value; }
    }
}

```

Fim do código

Vídeo de explicação do Sistema

<https://youtu.be/fK35gowOPRQ>



Arquivos disponíveis no GitHub

https://github.com/Luciano-Lageano/SsistemaComputacional-Hands_VI.git

Considerações

Foi usado o visual studio por ser um programação orientada objeto, e o banco de dados o MySql por ser gratuito.

Este sistema será implementado para produção real após alguns testes de execução.

Referências Bibliográficas

https://www.youtube.com/watch?v=_q3j25ACmQ4&list=PLzFwmaV5qBRTahxAgtZ8hxLyE4Tf0Ji2m&index=1&pp=gAQBiAQB&ab_channel=HashtagPrograma%C3%A7%C3%A3o (CRUD em Python)

https://www.youtube.com/watch?v=oyO_tcR3KaU&list=PLzFwmaV5qBRTahxAgtZ8hxLyE4Tf0Ji2m&index=5&ab_channel=TomasEdson (Botão Next, Previous, Primeiro e Último Visual Studio e Banco MYSQL)

https://www.youtube.com/watch?v=qMIJy2qRC_4&list=PLzFwmaV5qBRTahxAgtZ8hxLyE4Tf0Ji2m&index=6&ab_channel=Rogers (conexão com MySql com o Windows Forms)

https://www.youtube.com/watch?v=2xM8Mhh7Plw&list=PLzFwmaV5qBRTahxAgtZ8hxLyE4Tf0Ji2m&index=8&ab_channel=LuizmarOliveiradasNeves (abrir outro formulário)