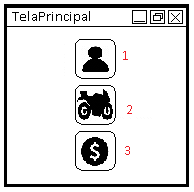
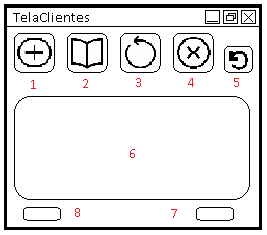
|  |
| --- |
| **Turma:** T10 – Join |
| **Professor:** Oswaldo Borges Peres |
| **Equipe:** T10Junior |
| **Componentes:** Luciano Juliano Dutra Escobar |

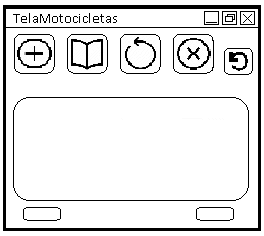
Esta é a tela principal do programa, a única interação do usuário é com as três “btn” ilustradas abaixo, seriam opções de acesso para clientes (1), motocicletas (2) ou vendas (3).



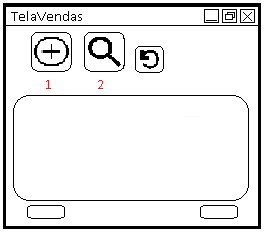
Clicando na tela de clientes, o usuário irá para a “TelaClientes”, não qual existem as opções de acesso para criar (1), consultar (2), atualizar (3) e deletar (4) um cadastro referente a clientes. Conforme o usuário clica no “btn” desejado, o espaço em branco (6) apresenta as informações necessárias para a opção escolhida e conforme o preenchimento dos dados vão sendo inseridos, podem ser atualizados (7) ou cancelados (8) nos últimos dois botões. Além destes, existe também um “btn” para retornar (5) a “TelaPrincipal”.



A tela motocicletas é idêntica a tela de clientes, porém com as informações pertinentes as motos.



E por último e não menos importante, a tela de vendas possui dois botões não apresentados ainda, que seriam o de cadastrar uma nova venda (1) e o de pesquisar uma determinada venda. O funcionamento dos “buttons” é o mesmo das telas de cliente e motocicletas.



**CÓDIGO GERADO NO MYSQL**

create database crud;

create table clientes(

ID\_cliente int not null auto\_increment primary key,

nome varchar(40),

cpf int,

endereco varchar(100),

email varchar(100),

idade int);

describe clientes;

UPDATE CLIENTES SET NOME = "JOAO CARLOS AUGUSTO", IDADE = 25 WHERE ID\_cliente = 2;

select \* from clientes;

create table motocicletas(

ID\_moto int not null auto\_increment primary key,

marca varchar(20),

modelo varchar(20),

cor varchar(20),

cilindrada int,

ano int,

preco double);

describe motocicletas;

select \* from motocicletas;

create table vendas(

codVenda int not null auto\_increment primary key,

preco double,

dataVenda date,

obs varchar(200),

ID\_cliente int,

ID\_moto int,

constraint fk\_cliVenda foreign key(ID\_cliente) references clientes (ID\_cliente),

constraint fk\_motoVenda foreign key (ID\_moto) references motocicletas (ID\_moto));

describe vendas;

select \* from vendas;

**CÓDIGO GERADO NO ECLIPSE**

**CLASSE PRINCIPAL**

package br.com.crud;

import java.util.Date;

import br.com.crud.dao.CrudDao;

public class Principal {

public static void main(String[] args) {

CrudDao crud = new CrudDao();

//Clientes cliente = new Clientes();

//Motocicletas moto = new Motocicletas();

Vendas venda = new Vendas();

/\*

cliente.setNome("Malu");

cliente.setCpf(5);

cliente.setEndereco("poa, 86");

cliente.setEmail("filhota@gmail.com");

cliente.setIdade(2);

crud.createCliente(cliente);

\*/

//Vendas venda = new Vendas();

//venda.setData(new Date());

/\*

for(Clientes lista : crud.getListaClientes()) {

System.out.println("Nome: "+lista.getNome());

System.out.println("CPF: "+lista.getCpf());

System.out.println("Endereco: "+lista.getEndereco());

System.out.println("Email: "+lista.getEmail());

System.out.println("Idade: "+lista.getIdade());

System.out.println("");

System.out.println("----\*\*----");

System.out.println("");

}\*/

/\*

cliente.setNome("Carlos Macedo");

cliente.setCpf(9999);

cliente.setEndereco("Rua Jaime");

cliente.setEmail("contapessoal@hotmail.com");

cliente.setIdade(48);

cliente.setIdCliente(4);

crud.updateCliente(cliente);

\*/

/\*

cliente.setIdCliente(4);

crud.deleteCliente(cliente);

\*/

/\*

moto.setMarca("Kawasaki");

moto.setModelo("ZX6R");

moto.setCor("Preta");

moto.setCilindrada(636);

moto.setAno(2025);

moto.setPreco(56000.00);

crud.createMoto(moto);

\*/

/\*

for(Motocicletas lista : crud.getListaMotos()) {

System.out.println("Marca: "+lista.getMarca());

System.out.println("Modelo: "+lista.getModelo());

System.out.println("Cor: "+lista.getCor());

System.out.println("Cilindrada: "+lista.getCilindrada());

System.out.println("Ano: "+lista.getAno());

System.out.println("Preco: "+lista.getPreco());

System.out.println("");

System.out.println("----\*\*----");

System.out.println("");

}

\*/

/\*

moto.setMarca("Kawasaki");

moto.setModelo("ZX10R");

moto.setCor("Branca");

moto.setCilindrada(636);

moto.setAno(2025);

moto.setPreco(56000.43);

moto.setIdMoto(2);

crud.updateMoto(moto);

\*/

/\*

moto.setIdMoto(2);

crud.deleteMoto(moto);

\*/

/\*

venda.setPreco(12000);

venda.setDataVenda(new Date());

venda.setObs("Testando o campo de observacoes no banco de dados..");

venda.setIdCliente(3);

venda.setIdMoto(1);

crud.createVendas(venda);

\*/

/\*

for(Vendas lista : crud.getListaVendas()) {

System.out.println("Cod. Venda: "+lista.getCodVenda());

System.out.println("Preco: "+lista.getPreco());

System.out.println("Data Venda: "+lista.getDataVenda());

System.out.println("Observacao: "+lista.getObs());

System.out.println("ID. Cliente: "+lista.getIdCliente());

System.out.println("ID. Moto: "+lista.getIdMoto());

System.out.println("");

System.out.println("----\*\*----");

System.out.println("");

}

\*/

}

**CLASSE CLIENTES**

package br.com.crud;

public class Clientes extends Interagir{

private int idCliente, cpf, idade;

private String nome, endereco, email;

public int getIdCliente() {

return idCliente;

}

public void setIdCliente(int idCliente) {

this.idCliente = idCliente;

}

public int getCpf() {

return cpf;

}

public void setCpf(int cpf) {

this.cpf = cpf;

}

public String getNome() {

return nome;

}

public void setNome(String nome) {

this.nome = nome;

}

public String getEndereco() {

return endereco;

}

public void setEndereco(String endereco) {

this.endereco = endereco;

}

public String getEmail() {

return email;

}

public void setEmail(String email) {

this.email = email;

}

public int getIdade() {

return idade;

}

public void setIdade(int idade) {

this.idade = idade;

}

}

**CLASSE MOTOCICLETAS**

package br.com.crud;

public class Motocicletas{

int idMoto, cilindrada, ano;

String marca, modelo, cor;

double preco;

public int getIdMoto() {

return idMoto;

}

public void setIdMoto(int idMoto) {

this.idMoto = idMoto;

}

public int getCilindrada() {

return cilindrada;

}

public void setCilindrada(int cilindrada) {

this.cilindrada = cilindrada;

}

public int getAno() {

return ano;

}

public void setAno(int ano) {

this.ano = ano;

}

public String getMarca() {

return marca;

}

public void setMarca(String marca) {

this.marca = marca;

}

public String getModelo() {

return modelo;

}

public void setModelo(String modelo) {

this.modelo = modelo;

}

public String getCor() {

return cor;

}

public void setCor(String cor) {

this.cor = cor;

}

public double getPreco() {

return preco;

}

public void setPreco(double preco) {

this.preco = preco;

}

}

**CLASSE VENDAS**

package br.com.crud;

import java.util.Date;

public class Vendas{

private int idCliente, idMoto, codVenda;

private double preco;

private String obs;

private Date dataVenda;

public int getIdCliente() {

return idCliente;

}

public void setIdCliente(int idCliente) {

this.idCliente = idCliente;

}

public int getIdMoto() {

return idMoto;

}

public void setIdMoto(int idMoto) {

this.idMoto = idMoto;

}

public int getCodVenda() {

return codVenda;

}

public void setCodVenda(int codVenda) {

this.codVenda = codVenda;

}

public double getPreco() {

return preco;

}

public void setPreco(double preco) {

this.preco = preco;

}

public Date getDataVenda() {

return dataVenda;

}

public void setDataVenda(Date dataVenda) {

this.dataVenda = dataVenda;

}

public String getObs() {

return obs;

}

public void setObs(String obs) {

this.obs = obs;

}

}

**CLASSE CONNECTIONFACTORY**

package br.com.crud.factory;

import java.sql.Connection;

import java.sql.DriverManager;

public class ConnectionFactory {

private static final String ***userName*** = "root";

private static final String ***password*** = "";

private static final String ***db\_url*** = "jdbc:mysql://localhost:3306/crud";

public static Connection createConnectionToMySQL() throws Exception {

//FAZ COM QUE A CLASSE SEJA CARREGADA PELA JVM

Class.*forName*("com.mysql.cj.jdbc.Driver");

//CRIA A CONEXÃO COM O DB

Connection connection = DriverManager.*getConnection*(***db\_url***, ***userName***, ***password***);

return connection;

}

public static void main(String[] args) throws Exception {

//RECUPERAR UMA CONEXÃO COM O DB

Connection con = *createConnectionToMySQL*();

//TESTAR SE A CONEXÃO É NULA

if(con!=null) {

System.***out***.println("Conexão realizada com sucesso!");

con.close();

}

}

}

**CLASSE CRUDDAO**

package br.com.crud.dao;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import br.com.crud.Clientes;

import br.com.crud.Motocicletas;

import br.com.crud.Vendas;

import br.com.crud.factory.ConnectionFactory;

public class CrudDao {

public void createCliente(Clientes cliente) {

String sql = "insert into clientes (nome, cpf, endereco, email, idade) values (?, ?, ?, ?, ?)";

Connection conn = null;

PreparedStatement pstm = null;

try {

//CRIAR UMA CONEXÃO COM O DB

conn = ConnectionFactory.createConnectionToMySQL();

//EXECUTA UMA QUERY

pstm = conn.prepareStatement(sql);

//ADICIONAR OS VALORES ESPERADOS PELA QUERY

pstm.setString(1, cliente.getNome());

pstm.setInt(2, cliente.getCpf());

pstm.setString(3, cliente.getEndereco());

pstm.setString(4, cliente.getEmail());

pstm.setInt(5, cliente.getIdade());

//EXECUTA A QUERY

pstm.execute();

System.out.println("Cliente cadastrado com sucesso!");

} catch (Exception e) {

e.printStackTrace();

//FINALIZA AS CONEXÕES

}finally {

try {

if(pstm!=null){

pstm.close();

}

if(conn!=null) {

conn.close();

}

}catch (Exception e) {

e.printStackTrace();

}

}

}

public void updateCliente(Clientes cliente) {

String sql = "update clientes set nome = ?, cpf = ?, endereco = ?, email = ?, idade = ? where ID\_cliente = ?";

Connection conn = null;

PreparedStatement pstm = null;

try {

conn = ConnectionFactory.createConnectionToMySQL();

pstm = conn.prepareStatement(sql);

pstm.setString(1, cliente.getNome());

pstm.setInt(2, cliente.getCpf());

pstm.setString(3, cliente.getEndereco());

pstm.setString(4, cliente.getEmail());

pstm.setInt(5, cliente.getIdade());

pstm.setInt(6, cliente.getIdCliente());

//pstm.setDate(3, new Date(cliente.getDataVenda().getTime()));

pstm.execute();

System.out.println("Contato editado com sucesso!");

}catch (Exception e) {

e.printStackTrace();

}finally {

try{

if(pstm!=null){

pstm.close();

}

if(conn!=null) {

conn.close();

}

}catch (Exception e) {

e.printStackTrace();

}

}

}

public List<Clientes> getListaClientes(){

String sql = "select \* from clientes";

List<Clientes> listaClientes = new ArrayList<Clientes>();

Connection conn = null;

PreparedStatement pstm = null;

// CLASSE QUE RECUPERA OS DADOS DO DB

ResultSet rst = null;

try {

conn = ConnectionFactory.createConnectionToMySQL();

pstm = conn.prepareStatement(sql);

rst = pstm.executeQuery();

while(rst.next()) {

Clientes lista = new Clientes();

lista.setNome(rst.getString("nome"));

lista.setCpf(rst.getInt("cpf"));

lista.setEndereco(rst.getString("endereco"));

lista.setEmail(rst.getString("email"));

lista.setIdade(rst.getInt("idade"));

listaClientes.add(lista);

}

}catch (Exception e) {

e.printStackTrace();

}finally {

try{

if(rst!=null){

rst.close();

}

if(pstm!=null) {

pstm.close();

}

if(conn!=null) {

conn.close();

}

}catch (Exception e) {

e.printStackTrace();

}

}

return listaClientes;

}

public void deleteCliente(Clientes id) {

String sql = "delete from clientes where ID\_cliente = ?";

Connection conn = null;

PreparedStatement pstm = null;

try {

conn = ConnectionFactory.createConnectionToMySQL();

pstm = conn.prepareStatement(sql);

pstm.setInt(1, id.getIdCliente());

pstm.execute();

System.out.println("Contato excluído com sucesso! ");

}catch (Exception e) {

e.printStackTrace();

}finally {

try{

if(pstm!=null){

pstm.close();

}

if(conn!=null) {

conn.close();

}

}catch (Exception e) {

e.printStackTrace();

}

}

}

public void createMoto(Motocicletas moto) {

String sql = "insert into motocicletas (marca, modelo, cor, cilindrada, ano, preco) values (?, ?, ?, ?, ?, ?)";

Connection conn = null;

PreparedStatement pstm = null;

try {

//CRIAR UMA CONEXÃO COM O DB

conn = ConnectionFactory.createConnectionToMySQL();

//EXECUTA UMA QUERY

pstm = conn.prepareStatement(sql);

//ADICIONAR OS VALORES ESPERADOS PELA QUERY

pstm.setString(1, moto.getMarca());

pstm.setString(2, moto.getModelo());

pstm.setString(3, moto.getCor());

pstm.setInt(4, moto.getCilindrada());

pstm.setInt(5, moto.getAno());

pstm.setDouble(6, moto.getPreco());

//EXECUTA A QUERY

pstm.execute();

System.out.println("Moto cadastrada com sucesso!");

} catch (Exception e) {

e.printStackTrace();

//FINALIZA AS CONEXÕES

}finally {

try {

if(pstm!=null){

pstm.close();

}

if(conn!=null) {

conn.close();

}

}catch (Exception e) {

e.printStackTrace();

}

}

}

public void updateMoto(Motocicletas moto) {

String sql = "update motocicletas set marca = ?, modelo = ?, cor = ?, cilindrada = ?, ano = ?, preco = ? where ID\_moto = ?";

Connection conn = null;

PreparedStatement pstm = null;

try {

conn = ConnectionFactory.createConnectionToMySQL();

pstm = conn.prepareStatement(sql);

pstm.setString(1, moto.getMarca());

pstm.setString(2, moto.getModelo());

pstm.setString(3, moto.getCor());

pstm.setInt(4, moto.getCilindrada());

pstm.setInt(5, moto.getAno());

pstm.setDouble(6, moto.getPreco());

pstm.setInt(7, moto.getIdMoto());

//pstm.setDate(3, new Date(cliente.getDataVenda().getTime()));

pstm.execute();

System.out.println("Moto editada com sucesso!");

}catch (Exception e) {

e.printStackTrace();

}finally {

try{

if(pstm!=null){

pstm.close();

}

if(conn!=null) {

conn.close();

}

}catch (Exception e) {

e.printStackTrace();

}

}

}

public List<Motocicletas> getListaMotos(){

String sql = "select \* from motocicletas";

List<Motocicletas> listaMotos = new ArrayList<Motocicletas>();

Connection conn = null;

PreparedStatement pstm = null;

// CLASSE QUE RECUPERA OS DADOS DO DB

ResultSet rst = null;

try {

conn = ConnectionFactory.createConnectionToMySQL();

pstm = conn.prepareStatement(sql);

rst = pstm.executeQuery();

while(rst.next()) {

Motocicletas lista = new Motocicletas();

lista.setMarca(rst.getString("marca"));

lista.setModelo(rst.getString("modelo"));

lista.setCor(rst.getString("cor"));

lista.setCilindrada(rst.getInt("cilindrada"));

lista.setAno(rst.getInt("ano"));

lista.setPreco(rst.getDouble("preco"));

listaMotos.add(lista);

}

}catch (Exception e) {

e.printStackTrace();

}finally {

try{

if(rst!=null){

rst.close();

}

if(pstm!=null) {

pstm.close();

}

if(conn!=null) {

conn.close();

}

}catch (Exception e) {

e.printStackTrace();

}

}

return listaMotos;

}

public void deleteMoto(Motocicletas id) {

String sql = "delete from motocicletas where ID\_moto = ?";

Connection conn = null;

PreparedStatement pstm = null;

try {

conn = ConnectionFactory.createConnectionToMySQL();

pstm = conn.prepareStatement(sql);

pstm.setInt(1, id.getIdMoto());

pstm.execute();

System.out.println("Moto excluída com sucesso! ");

}catch (Exception e) {

e.printStackTrace();

}finally {

try{

if(pstm!=null){

pstm.close();

}

if(conn!=null) {

conn.close();

}

}catch (Exception e) {

e.printStackTrace();

}

}

}

public void createVendas(Vendas venda) {

String sql = "insert into vendas (preco, dataVenda, obs, ID\_cliente, ID\_moto) values (?, ?, ?, ?, ?)";

Connection conn = null;

PreparedStatement pstm = null;

try {

//CRIAR UMA CONEXÃO COM O DB

conn = ConnectionFactory.createConnectionToMySQL();

//EXECUTA UMA QUERY

pstm = conn.prepareStatement(sql);

//ADICIONAR OS VALORES ESPERADOS PELA QUERY

pstm.setDouble(1, venda.getPreco());

pstm.setDate(2, new java.sql.Date(venda.getDataVenda().getTime()));

pstm.setString(3, venda.getObs());

pstm.setInt(4, venda.getIdCliente());

pstm.setInt(5, venda.getIdMoto());

//EXECUTA A QUERY

pstm.execute();

System.out.println("Venda cadastrada com sucesso!");

} catch (Exception e) {

e.printStackTrace();

//FINALIZA AS CONEXÕES

}finally {

try {

if(pstm!=null){

pstm.close();

}

if(conn!=null) {

conn.close();

}

}catch (Exception e) {

e.printStackTrace();

}

}

}

public List<Vendas> getListaVendas(){

String sql = "select \* from vendas";

List<Vendas> listaVendas = new ArrayList<Vendas>();

Connection conn = null;

PreparedStatement pstm = null;

// CLASSE QUE RECUPERA OS DADOS DO DB

ResultSet rst = null;

try {

conn = ConnectionFactory.createConnectionToMySQL();

pstm = conn.prepareStatement(sql);

rst = pstm.executeQuery();

while(rst.next()) {

Vendas lista = new Vendas();

lista.setCodVenda(rst.getInt("CodVenda"));

lista.setPreco(rst.getDouble("Preco"));

lista.setDataVenda(rst.getDate("DataVenda"));

lista.setObs(rst.getString("Obs"));

lista.setIdCliente(rst.getInt("Id\_Cliente"));

lista.setIdMoto(rst.getInt("Id\_Moto"));

listaVendas.add(lista);

}

}catch (Exception e) {

e.printStackTrace();

}finally {

try{

if(rst!=null){

rst.close();

}

if(pstm!=null) {

pstm.close();

}

if(conn!=null) {

conn.close();

}

}catch (Exception e) {

e.printStackTrace();

}

}

return listaVendas;

}

}