Trabalho Final

Programação de Soluções Computacionais

Nome: Joao Vitor Guimaraes

Santos

RA: 972312195

----------------------------------------

Nome: Pedro Henrique

Antero Pereira

RA: 972311524

----------------------------------------

Faseh

Curso: Analise de desenvolvimento de sistemas

Turno: Noite

CRIAÇAO DA CLASSE MAIN:

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.Scanner;

public class Main {

static Connection connection = null;

static Scanner ler = new Scanner(System.in);

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

int linha;

do {

linha = exibirMenuPrincipal();

System.out.println();

switch (linha) {

case 1:

menuGarcom();

break;

case 2:

menuMesa();

break;

case 0:

System.out.println("Finalizado");

break;

default:

System.out.println("ERRO!");

break;

}

System.out.println();

} while (linha != 0);

}

private static int exibirMenuPrincipal() {

System.out.println("Menu Principal:");

System.out.println("1) Gerenciar Garçons");

System.out.println("2) Gerenciar Mesas");

System.out.println("Digite 0 para Finalizar");

return ler.nextInt();

}

private static void menuGarcom() throws Exception {

int linha;

do {

linha = exibirMenuGarcom();

System.out.println();

switch (linha) {

case 1:

inserirGarcom();

break;

case 2:

removerGarcomPeloEmail();

break;

case 3:

buscarGarcomPeloEmail();

break;

case 4:

alterarGarcom();

break;

case 5:

calcularMediaSalarioFixo();

break;

case 0:

System.out.println("Voltando ao Menu Principal");

break;

default:

System.out.println("ERRO!");

break;

}

System.out.println();

} while (linha != 0);

}

private static int exibirMenuGarcom() {

System.out.println("Menu Garçom:");

System.out.println("1) Inserir Garçom");

System.out.println("2) Remover Garçom");

System.out.println("3) Buscar Garçom");

System.out.println("4) Alterar Garçom");

System.out.println("5) Calcular Média do Salário Fixo");

System.out.println("Digite 0 para Voltar ao Menu Principal");

return ler.nextInt();

}

private static void menuMesa() throws Exception {

int linha;

do {

linha = exibirMenuMesa();

System.out.println();

switch (linha) {

case 1:

inserirMesa();

break;

case 2:

removerMesaPeloNumero();

break;

case 3:

buscarMesaPeloNumero();

break;

case 4:

buscarGarcomResponsavelPelaMesa();

break;

case 5:

buscarMesasLivresComGarcomResponsavel();

break;

case 6:

buscarMesasOcupadasPorGarcom();

break;

case 7:

calcularQuantidadeMesasAtendidasPorGarcom();

break;

case 0:

System.out.println("Voltando ao Menu Principal");

break;

default:

System.out.println("ERRO!");

break;

}

System.out.println();

} while (linha != 0);

}

private static int exibirMenuMesa() {

System.out.println("Menu Mesa:");

System.out.println("1) Inserir Mesa");

System.out.println("2) Remover Mesa");

System.out.println("3) Buscar Mesa");

System.out.println("4) Buscar Garçom Responsável pela Mesa");

System.out.println("5) Buscar Mesas Livres com Garçom Responsável");

System.out.println("6) Buscar Mesas Ocupadas por Garçom");

System.out.println("7) Calcular Quantidade de Mesas Atendidas por Garçom");

System.out.println("Digite 0 para Voltar ao Menu Principal");

return ler.nextInt();

}

// CALCULAR MEDIA

private static void calcularMediaSalarioFixo() throws SQLException {

try {

connection = ConexaoDB.getInstance();

String sql = "SELECT AVG(salarioFixo) AS mediaSalarioFixo FROM garcom";

PreparedStatement stmt = connection.prepareStatement(sql);

ResultSet resultado = stmt.executeQuery();

if (resultado.next()) {

double mediaSalarioFixo = resultado.getDouble("mediaSalarioFixo");

System.out.println("Média de Salário: " + mediaSalarioFixo);

}

resultado.close();

stmt.close();

} catch (Exception e) {

System.out.println("Erro no cálculo da Média!");

e.printStackTrace();

}

}

// INSERIR GARÇOM

private static void inserirGarcom() throws SQLException {

try {

connection = ConexaoDB.getInstance();

String sql = "INSERT INTO garcom (id\_garcom, nome, cpf, dataNascimento, email, telefone, sexo, salarioFixo) VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

PreparedStatement stmt = connection.prepareStatement(sql);

System.out.println("Informe o ID do garçom: ");

int id\_garcom = ler.nextInt();

ler.nextLine();

System.out.println("Informe o Nome do garçom: ");

String nome = ler.nextLine();

System.out.println("Informe o CPF do garçom: ");

String cpf = ler.nextLine();

System.out.println("Informe a data de nascimento do garçom: ");

String dataNascimento = ler.nextLine();

System.out.println("Informe o email do garçom: ");

String email = ler.nextLine();

System.out.println("Informe o telefone do garçom: ");

String telefone = ler.nextLine();

System.out.println("Informe o sexo do garçom: ");

String sexo = ler.nextLine();

System.out.println("Informe o salário fixo do garçom: ");

double salarioFixo = ler.nextDouble();

stmt.setInt(1, id\_garcom);

stmt.setString(2, nome);

stmt.setString(3, cpf);

stmt.setString(4, dataNascimento);

stmt.setString(5, email);

stmt.setString(6, telefone);

stmt.setString(7, sexo);

stmt.setDouble(8, salarioFixo);

stmt.executeUpdate();

stmt.close();

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

} catch (Exception e) {

System.out.println("Não foi possível cadastrar o Garçom");

e.printStackTrace();

}

}

// REMOVER GARÇOM

public static void removerGarcomPeloEmail() throws Exception {

connection = ConexaoDB.getInstance();

String sql = "DELETE FROM garcom WHERE email LIKE ?";

System.out.println("Informe o email do garçom que deseja remover: ");

String emailDoGarcomQueSeraRemovido = ler.nextLine();

PreparedStatement stmt = connection.prepareStatement(sql);

stmt.setString(1, emailDoGarcomQueSeraRemovido);

stmt.execute();

stmt.close();

}

// BUSCAR GARÇOM

public static void buscarGarcomPeloEmail() throws Exception {

connection = ConexaoDB.getInstance();

String sql = "SELECT \* FROM garcom WHERE email LIKE ?";

System.out.println("Informe o email do garçom que deseja buscar: ");

String emailBuscado = ler.nextLine();

PreparedStatement stmt = connection.prepareStatement(sql);

stmt.setString(1, emailBuscado);

ResultSet resultado = stmt.executeQuery();

if (resultado.next()) {

int id\_garcom = resultado.getInt("id\_garcom");

String nome = resultado.getString("nome");

String cpf = resultado.getString("cpf");

String dataNascimento = resultado.getString("dataNascimento");

String email = resultado.getString("email");

String telefone = resultado.getString("telefone");

String sexo = resultado.getString("sexo");

double salarioFixo = resultado.getDouble("salarioFixo");

System.out.println("Garçom encontrado:");

System.out.println("ID: " + id\_garcom);

System.out.println("Nome: " + nome);

System.out.println("CPF: " + cpf);

System.out.println("Data de Nascimento: " + dataNascimento);

System.out.println("Email: " + email);

System.out.println("Telefone: " + telefone);

System.out.println("Sexo: " + sexo);

System.out.println("Salário Fixo: " + salarioFixo);

} else {

System.out.println("Garçom não encontrado!");

}

resultado.close();

stmt.close();

}

// ALTERAR GARÇOM

public static void alterarGarcom() throws Exception {

connection = ConexaoDB.getInstance();

System.out.println("Digite o email do garçom que deseja alterar: ");

String emailGarcomAlterar = ler.nextLine();

String sql = "SELECT \* FROM garcom WHERE email LIKE ?";

PreparedStatement stmt = connection.prepareStatement(sql);

stmt.setString(1, emailGarcomAlterar);

ResultSet resultado = stmt.executeQuery();

if (resultado.next()) {

int id\_garcom = resultado.getInt("id\_garcom");

String nome = resultado.getString("nome");

String cpf = resultado.getString("cpf");

String dataNascimento = resultado.getString("dataNascimento");

String email = resultado.getString("email");

String telefone = resultado.getString("telefone");

String sexo = resultado.getString("sexo");

double salarioFixo = resultado.getDouble("salarioFixo");

System.out.println("Informe o novo nome do garçom: ");

nome = ler.nextLine();

System.out.println("Informe o novo CPF do garçom: ");

cpf = ler.nextLine();

System.out.println("Informe a nova data de nascimento do garçom: ");

dataNascimento = ler.nextLine();

System.out.println("Informe o novo telefone do garçom: ");

telefone = ler.nextLine();

System.out.println("Informe o novo sexo do garçom: ");

sexo = ler.nextLine();

System.out.println("Informe o novo email:");

email = ler.nextLine();

System.out.println("Informe o novo salário fixo do garçom: ");

salarioFixo = ler.nextDouble();

sql = "UPDATE garcom SET nome = ?, cpf = ?, dataNascimento = ?, telefone = ?, sexo = ?, email = ?, salarioFixo = ? WHERE id\_garcom = ?";

stmt = connection.prepareStatement(sql);

stmt.setString(1, nome);

stmt.setString(2, cpf);

stmt.setString(3, dataNascimento);

stmt.setString(4, telefone);

stmt.setString(5, sexo);

stmt.setString(6, email);

stmt.setDouble(7, salarioFixo);

stmt.setInt(8, id\_garcom);

stmt.executeUpdate();

System.out.println("Garçom alterado com sucesso!");

} else {

System.out.println("Garçom não encontrado!");

}

resultado.close();

stmt.close();

}

// INSERIR MESA

public static void inserirMesa() throws SQLException {

try {

connection = ConexaoDB.getInstance();

String sql = "INSERT INTO mesa (numero\_mesa, situacao, capacidade\_maxima) VALUES (?, ?, ?)";

PreparedStatement stmt = connection.prepareStatement(sql);

System.out.println("Informe o número da mesa: ");

int numeroMesa = ler.nextInt();

ler.nextLine();

System.out.println("Informe a situação da mesa (livre, ocupada, reservada): ");

String situacao = ler.nextLine();

System.out.println("Informe a capacidade máxima da mesa: ");

int capacidadeMaxima = ler.nextInt();

stmt.setInt(1, numeroMesa);

stmt.setString(2, situacao);

stmt.setInt(3, capacidadeMaxima);

stmt.executeUpdate();

stmt.close();

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

} catch (Exception e) {

System.out.println("Não foi possível cadastrar a mesa");

e.printStackTrace();

}

}

// REMOVER MESA

public static void removerMesaPeloNumero() throws Exception {

connection = ConexaoDB.getInstance();

String sql = "DELETE FROM mesa WHERE numero\_mesa = ?";

System.out.println("Informe o número da mesa que deseja remover: ");

int numeroMesa = ler.nextInt();

PreparedStatement stmt = connection.prepareStatement(sql);

stmt.setInt(1, numeroMesa);

stmt.execute();

stmt.close();

}

// BUSCAR MESA

public static void buscarMesaPeloNumero() throws Exception {

connection = ConexaoDB.getInstance();

String sql = "SELECT \* FROM mesa WHERE numero\_mesa = ?";

System.out.println("Informe o número da mesa que deseja buscar: ");

int numeroMesa = ler.nextInt();

PreparedStatement stmt = connection.prepareStatement(sql);

stmt.setInt(1, numeroMesa);

ResultSet resultado = stmt.executeQuery();

if (resultado.next()) {

int numeroMesaEncontrada = resultado.getInt("numero\_mesa");

String situacao = resultado.getString("situacao");

int capacidadeMaxima = resultado.getInt("capacidade\_maxima");

System.out.println("Mesa encontrada:");

System.out.println("Número da Mesa: " + numeroMesaEncontrada);

System.out.println("Situação: " + situacao);

System.out.println("Capacidade Máxima: " + capacidadeMaxima);

} else {

System.out.println("Mesa não encontrada!");

}

resultado.close();

stmt.close();

}

// BUSCAR GARÇOM RESPONSÁVEL PELA MESA

public static void buscarGarcomResponsavelPelaMesa() throws Exception {

connection = ConexaoDB.getInstance();

String sql = "SELECT garcom.\* FROM garcom JOIN mesa ON garcom.id\_garcom = mesa.garcom\_id WHERE mesa.numero\_mesa = ?";

System.out.println("Informe o número da mesa: ");

int numeroMesa = ler.nextInt();

PreparedStatement stmt = connection.prepareStatement(sql);

stmt.setInt(1, numeroMesa);

ResultSet resultado = stmt.executeQuery();

if (resultado.next()) {

int id\_garcom = resultado.getInt("id\_garcom");

String nome = resultado.getString("nome");

String cpf = resultado.getString("cpf");

String dataNascimento = resultado.getString("dataNascimento");

String email = resultado.getString("email");

String telefone = resultado.getString("telefone");

String sexo = resultado.getString("sexo");

double salarioFixo = resultado.getDouble("salarioFixo");

System.out.println("Garçom responsável pela mesa:");

System.out.println("ID: " + id\_garcom);

System.out.println("Nome: " + nome);

System.out.println("CPF: " + cpf);

System.out.println("Data de Nascimento: " + dataNascimento);

System.out.println("Email: " + email);

System.out.println("Telefone: " + telefone);

System.out.println("Sexo: " + sexo);

System.out.println("Salário Fixo: " + salarioFixo);

} else {

System.out.println("Não há garçom responsável pela mesa!");

}

resultado.close();

stmt.close();

}

// BUSCAR MESAS LIVRES COM GARÇOM RESPONSÁVEL

public static void buscarMesasLivresComGarcomResponsavel() throws Exception {

connection = ConexaoDB.getInstance();

String sql = "SELECT mesa.\* FROM mesa JOIN garcom ON mesa.garcom\_id = garcom.id\_garcom WHERE mesa.situacao = 'livre'";

PreparedStatement stmt = connection.prepareStatement(sql);

ResultSet resultado = stmt.executeQuery();

if (resultado.next()) {

System.out.println("Mesas livres com garçom responsável:");

while (resultado.next()) {

int numeroMesa = resultado.getInt("numero\_mesa");

String situacao = resultado.getString("situacao");

int capacidadeMaxima = resultado.getInt("capacidade\_maxima");

System.out.println("Número da Mesa: " + numeroMesa);

System.out.println("Situação: " + situacao);

System.out.println("Capacidade Máxima: " + capacidadeMaxima);

System.out.println();

}

} else {

System.out.println("Não há mesas livres com garçom responsável!");

}

resultado.close();

stmt.close();

}

// BUSCAR MESAS OCUPADAS POR GARÇOM

public static void buscarMesasOcupadasPorGarcom() throws Exception {

connection = ConexaoDB.getInstance();

String sql = "SELECT mesa.\* FROM mesa JOIN garcom ON mesa.garcom\_id = garcom.id\_garcom WHERE mesa.situacao = 'ocupada'";

PreparedStatement stmt = connection.prepareStatement(sql);

ResultSet resultado = stmt.executeQuery();

if (resultado.next()) {

System.out.println("Mesas ocupadas por garçom:");

while (resultado.next()) {

int numeroMesa = resultado.getInt("numero\_mesa");

String situacao = resultado.getString("situacao");

int capacidadeMaxima = resultado.getInt("capacidade\_maxima");

System.out.println("Número da Mesa: " + numeroMesa);

System.out.println("Situação: " + situacao);

System.out.println("Capacidade Máxima: " + capacidadeMaxima);

System.out.println();

}

} else {

System.out.println("Não há mesas ocupadas por garçom!");

}

resultado.close();

stmt.close();

}

// CALCULAR QUANTIDADE DE MESAS ATENDIDAS POR GARÇOM

public static void calcularQuantidadeMesasAtendidasPorGarcom() throws Exception {

connection = ConexaoDB.getInstance();

String sql = "SELECT garcom.id\_garcom, garcom.nome, COUNT(mesa.numero\_mesa) AS quantidadeMesasAtendidas FROM garcom JOIN mesa ON garcom.id\_garcom = mesa.garcom\_id WHERE mesa.situacao = 'ocupada' GROUP BY garcom.id\_garcom";

PreparedStatement stmt = connection.prepareStatement(sql);

ResultSet resultado = stmt.executeQuery();

if (resultado.next()) {

System.out.println("Quantidade de mesas atendidas por garçom:");

while (resultado.next()) {

int idGarcom = resultado.getInt("id\_garcom");

String nomeGarcom = resultado.getString("nome");

int quantidadeMesasAtendidas = resultado.getInt("quantidadeMesasAtendidas");

System.out.println("ID do Garçom: " + idGarcom);

System.out.println("Nome do Garçom: " + nomeGarcom);

System.out.println("Quantidade de Mesas Atendidas: " + quantidadeMesasAtendidas);

System.out.println();

}

} else {

System.out.println("Não há mesas ocupadas por garçom!");

}

resultado.close();

stmt.close();

}

}

CRIANDO A CLASSE GARÇOM:

 public class Garcom {   
int id\_garcom;   
String nome;   
String cpf;   
String dataNascimento;   
String email;   
double telefone;   
String sexo;   
double salarioFixo;   
   
   
public Garcom() {   
   
}   
   
public Garcom(String nome, String cpf, String dataNascimento, String email, double telefone, String sexo, double salarioFixo, int id\_garcom) {   
this.nome = nome;   
this.cpf = cpf;   
this.dataNascimento = dataNascimento;   
this.email = email;   
this.telefone = telefone;   
this.sexo = sexo;   
this.salarioFixo = salarioFixo;   
this.id\_garcom = id\_garcom;   
}   
   
public Garcom(int idGarcom, String nome, String cpf, String dataNascimento, String email, double telefone, String sexo, double salarioFixo) {   
}   
   
   
public String getNome() {   
return nome;   
}   
   
public void setNome(String nome) {   
this.nome = nome;   
}   
   
public String getCpf() {   
return cpf;   
}   
   
public void setCpf(String cpf) {   
this.cpf = cpf;   
}   
   
public String getDataNascimento() {   
return dataNascimento;   
}   
   
public void setDataNascimento(String dataNascimento) {   
this.dataNascimento = dataNascimento;   
}   
   
public String getEmail() {   
return email;   
}   
   
public void setEmail(String email) {   
this.email = email;   
}   
   
public double getTelefone() {   
return telefone;   
}   
   
public void setTelefone(double telefone) {   
this.telefone = telefone;   
}   
   
public String getSexo() {   
return sexo;   
}   
   
public void setSexo(String sexo) {   
this.sexo = sexo;   
}   
   
public double getSalarioFixo() {   
return salarioFixo;   
}   
   
public void setSalarioFixo(double salarioFixo) {   
this.salarioFixo = salarioFixo;   
}   
   
public int getId\_garcom() {   
return id\_garcom;   
}   
   
public void setId\_garcom(int id\_garcom) {   
this.id\_garcom = id\_garcom;   
}   
}

CRIANDO A CLASSE MESA:

Class mesa:

public class Mesa {

private int numeroMesa;

private String situacao;

private int capacidadeMaxima;

private int garcomId;

Mesa(int numeroMesa, String situacao, int capacidadeMaxima, int garcomId) {

this.numeroMesa = numeroMesa;

this.situacao = situacao;

this.capacidadeMaxima = capacidadeMaxima;

this.garcomId = garcomId;

}

public int getNumeroMesa() {

return numeroMesa;

}

public void setNumeroMesa(int numeroMesa) {

this.numeroMesa = numeroMesa;

}

public String getSituacao() {

return situacao;

}

public void setSituacao(String situacao) {

this.situacao = situacao;

}

public int getCapacidadeMaxima() {

return capacidadeMaxima;

}

public void setCapacidadeMaxima(int capacidadeMaxima) {

this.capacidadeMaxima = capacidadeMaxima;

}

public int getGarcomId() {

return garcomId;

}

public void setGarcomId(int garcomId) {

this.garcomId = garcomId;

}

}

CRIANDO A CLASSE CONEXAODB:

import java.sql.Connection;   
import java.sql.DriverManager;   
import java.sql.SQLException;   
   
public class ConexaoDB {   
   
private static Connection *connection* = null;   
private String fonte = "Restaurante?useTimezone=true&serverTimezone=UTC";   
private String jdbcDriver = "com.mysql.cj.jdbc.Driver";   
private ConexaoDB () throws ClassNotFoundException {   
   
try {   
Class.*forName*(jdbcDriver);   
*connection* = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/" + fonte, "root", "1234");   
} catch (SQLException e) {   
e.printStackTrace();   
System.*out*.println("Ocorreu um erro ao conectar o banco de dados");   
}   
}   
   
public static Connection getInstance() throws ClassNotFoundException {   
if (*connection* == null){   
new ConexaoDB();   
}   
return *connection*;   
}   
   
}

 AVISOS!

Tenha em mente que para o código funcionar corretamente é necessário que o MySql e o  InteliJ estejam nas versões mais atualizadas. Sabendo disso é preciso baixar o conector do MySql que se encontra no link do github abaixo:

<https://github.com/PedroAnteroo/TrabalhoF.git>

Para não ocorrer duvidadas de como é o código, nesse Repositório também se encontram os mesmos códigos escritos acima na pasta “teste trabalho” junto com os códigos do MySql.

Tenha em mente que na classe conexaoDB sera necessario mudar a senha para a mesma do seu MySql:

*connection* = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/" + fonte, "root", "1234");

Onde está “1234”, coloque a sua senha do seu MySql para não ocorrer erros.

Para não ocorrer erros certifique-se que os seguintes comandos estejam no MySql:

CREATE TABLE mesa (

numero\_mesa INT PRIMARY KEY,

situacao ENUM('livre', 'ocupada', 'reservada'),

capacidade\_maxima INT,

garcom\_id INT,

FOREIGN KEY (garcom\_id) REFERENCES garcom (id\_garcom)

);

insert into mesa values ('2', 'livre', '4', '2');

select \* from mesa;