

```

package main.java.com.wfdai.weatherforecastdai.main;

import com.mysql.jdbc.jdbc2.optional.MysqlDataSource; import java.sql.Connection;
import java.sql.ResultSet; import java.sql.SQLException; import java.sql.Statement;
import main.java.com.wfdai.weatherforecastdai.main.weather.Weather; import
java.text.DateFormat; import java.util.Date; import java.text.SimpleDateFormat;
import java.time.LocalDate; import java.time.LocalDateTime; import
java.time.LocalTime; import java.util.ArrayList;

/** * Guarda e Recolhe Dados relativos a meteorologia na Base de Dados @au-
thor daniel */ public class Historico {

    DateFormat dateFormat = new SimpleDateFormat("yyyy/MM/dd HH:mm:ss");
    ArrayList<Integer> direcaoVento;
    ArrayList<Float> velocidadeVento;
    ArrayList<Integer> temperatura;
    ArrayList<Float> pressao;
    ArrayList<String> dataDados;
    ArrayList<Integer> humidade;
    ArrayList<Float> visibilidade;
    ArrayList<String> nascerSol;
    ArrayList<String> porSol;

    public Historico() {
        this.direcaoVento = new ArrayList<>();
        this.velocidadeVento = new ArrayList<>();
        this.temperatura = new ArrayList<>();
        this.pressao = new ArrayList<>();
        this.dataDados = new ArrayList<>();
        this.humidade = new ArrayList<>();
        this.visibilidade = new ArrayList<>();
        this.nascerSol = new ArrayList<>();
        this.porSol = new ArrayList<>();
    }

    /**
     * Recebe os dados e coloca-os na base de dados
     *
     * @param weather Objecto com os dados meteorologicos recolhidos.
     * @param localidade String com a localidade.
     */
    public void putHistorico(Weather weather, String localidade) {

        try {
            MysqlDataSource dataSource = new MysqlDataSource();
            DataBase database = new DataBase();
            dataSource.setUser(database.getUser());

```

```

        dataSource.setPassword(database.getPassword());
        dataSource.setServerName(database.getServerName());
        try (Connection conn = dataSource.getConnection()) {
            Statement st = conn.createStatement();
            st.executeUpdate("INSERT INTO mydb.Historico (`localidade`, `direcaoVento`, `velocidadeVento`, `temperatura`, `pressao`, `dataDados`, `humidade`, `visibilidade`, `nascerSol`, `porSol`)
                + "`pressao`, `dataDados`, `humidade`, `visibilidade`, `nascerSol`, `porSol`"
                + "VALUES ('" + localidade + "','" + weather.getDirecaoVento() + "','" + weather.getVelocidadeVento()
                + weather.getTemperatura() + "','" + weather.getPressao()
                + "','" + dateFormat.format(weather.getDataDados()) + "','" + weather.getHumidade()
                + "','" + weather.getVisibilidade() + "','" + weather.getNascerSol() + "','" + weather.getPorSol() + "')");
        }
    } catch (SQLException e) {
        System.err.println("Got an exception! ");
        System.err.println(e.getMessage());
    }
}

/**
 * Busca o histórico dos dados metereologicos relativamente a localidade
 * pretendida
 *
 * @param localidade String com a localidade.
 */
public void getHistorico(String localidade) {
    direcaoVento.clear();
    velocidadeVento.clear();
    temperatura.clear();
    pressao.clear();
    dataDados.clear();
    humidade.clear();
    visibilidade.clear();
    nascerSol.clear();
    porSol.clear();

    try {
        MysqlDataSource dataSource = new MysqlDataSource();
        DataBase database = new DataBase();
        dataSource.setUser(database.getUser());
        dataSource.setPassword(database.getPassword());
        dataSource.setServerName(database.getServerName());
        try (Connection conn = dataSource.getConnection()) {
            Statement st = conn.createStatement();
            ResultSet rs = st.executeQuery("
                + "SELECT max(`localidade`) as `localidade`, "
                + " max(`direcaoVento`) as `direcaoVento`, max(`velocidadeVento`) as `velocidadeVento`, "
                + " max(`temperatura`) as `temperatura`, max(`pressao`) as `pressao`, ma
            ");
        }
    }
}

```

```

        + " max(`humidade`) as `humidade`, max(`visibilidade`) as `visibilidade` "
        + " max(`porSol`) as `porSol` "
        + "FROM mydb.Historico "
        + "where localidade = '" + localidade + "' group by dataDados ;");

while (rs.next()) {
    direcaoVento.add(rs.getInt("direcaoVento"));
    velocidadeVento.add(rs.getFloat("velocidadeVento"));
    temperatura.add(rs.getInt("temperatura"));
    pressao.add(rs.getFloat("pressao"));

    LocalDate datePart = LocalDate.parse(rs.getDate("dataDados").toString());
    LocalTime timePart = LocalTime.parse(rs.getTime("dataDados").toString());
    LocalDateTime dt = LocalDateTime.of(datePart, timePart);
    Date data = java.sql.Timestamp.valueOf(dt);
    dataDados.add("\"" + data + "\"");
    humidade.add(rs.getInt("humidade"));
    visibilidade.add(rs.getFloat("visibilidade"));
    nascerSol.add("\"" + rs.getString("nascerSol") + "\"");
    porSol.add("\"" + rs.getString("porSol") + "\"");
    System.out.println("");
}
}
} catch (SQLException e) {
    System.err.println("Got an exception! ");
    System.err.println(e.getMessage());
}
}

public ArrayList<Integer> getDirecaoVento() {
    return direcaoVento;
}

public ArrayList<Float> getVelocidadeVento() {
    return velocidadeVento;
}

public ArrayList<Integer> getTemperatura() {
    return temperatura;
}

public ArrayList<Float> getPressao() {
    return pressao;
}

public ArrayList<String> getDataDados() {

```

```
        return dataDados;
    }

    public ArrayList<Integer> getHumidade() {
        return humidade;
    }

    public ArrayList<Float> getVisibilidade() {
        return visibilidade;
    }

    public ArrayList<String> getNascerSol() {
        return nascerSol;
    }

    public ArrayList<String> getPorSol() {
        return porSol;
    }
}
```