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 java.time.LocalDate; import java.time.LocalDateTime; import java.time.LocalTime; import java.util.ArrayList; import java.util.Date;

/** * Regista erros na Base de Dados e envia o registo para o broker @author daniel */ public class GestorErros {

```
String erro;
ArrayList<String> erros;
ArrayList<Date> time;

public GestorErros() {
    this.erros = new ArrayList<>();
    this.time = new ArrayList<>();
}

public GestorErros(String erro) {
    this.erro = erro;
}

Parser parser = new Parser();
Publisher publish = new Publisher();

/**
 * Recebe os erros e regista-os na BD
 *
 * @param erro String com a causa do erro
 */
public void putErro(String erro) {
    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();
            if (!erro.isEmpty()) {
                st.executeUpdate("INSERT INTO mydb.Erros (`erro`)"
                        + "VALUES ('" + erro + "')");
            }
        }

    } catch (SQLException e) {
```

```
        System.err.println("Got an exception! ");
        System.err.println(e.getMessage());
    }
}

/**
 * Recolhe os erros da BD e envia-os para o Broker
 *
 */
public void getErro() {
    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 * from mydb.Erros ");

            while (rs.next()) {
                erros.add(rs.getString("erro"));
                LocalDate datePart = LocalDate.parse(rs.getDate("time").toString());
                LocalTime timePart = LocalTime.parse(rs.getTime("time").toString());
                LocalDateTime dt = LocalDateTime.of(datePart, timePart);
                Date data = java.sql.Timestamp.valueOf(dt);
                time.add(data);

            }
        }
        parser.setParser(this);
        String mensagem = parser.getParsedMessage();
        publish.publish("/Erros", mensagem);
    } catch (SQLException e) {
        System.err.println("Got an exception! ");
        System.err.println(e.getMessage());
    }
}

public ArrayList<String> getErros() {
    return erros;
}

public ArrayList<Date> getTime() {
    return time;
}
```

```
public void setErro(String erro) {
    this.erro = erro;
}

}
```