# Universidade Federal Fluminense

## Ciência da Computação – Projeto de Banco de Dados

***Trabalho de implementação - Sistema de Gerenciamento de Processos Judiciais***

***Professor: Luís André***

***Aluno (Graduação): Gustavo Henrique Mello Dauer***

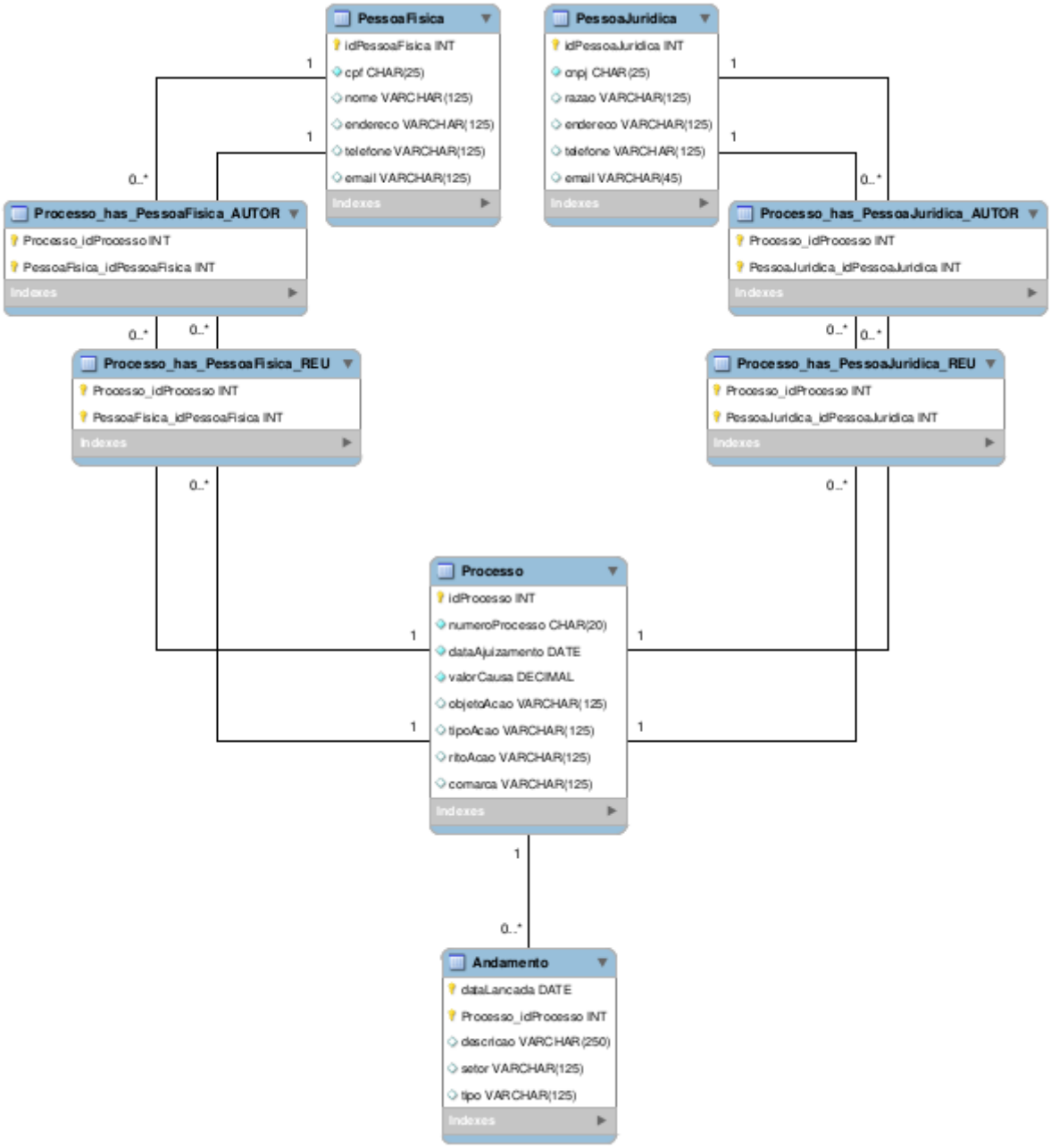
## Visão Geral do Sistema

O sistema deve gerenciar os processos judiciais do escritório. Cada processo pode ter vários autores e vários réus. Um processo tem diversos andamentos associados. Cada andamento só pode estar associado a um processo. Cada autor e réu do processo pode ser uma pessoa física ou uma pessoa jurídica. Um autor de um determinado processo não pode ser réu desse mesmo processo.

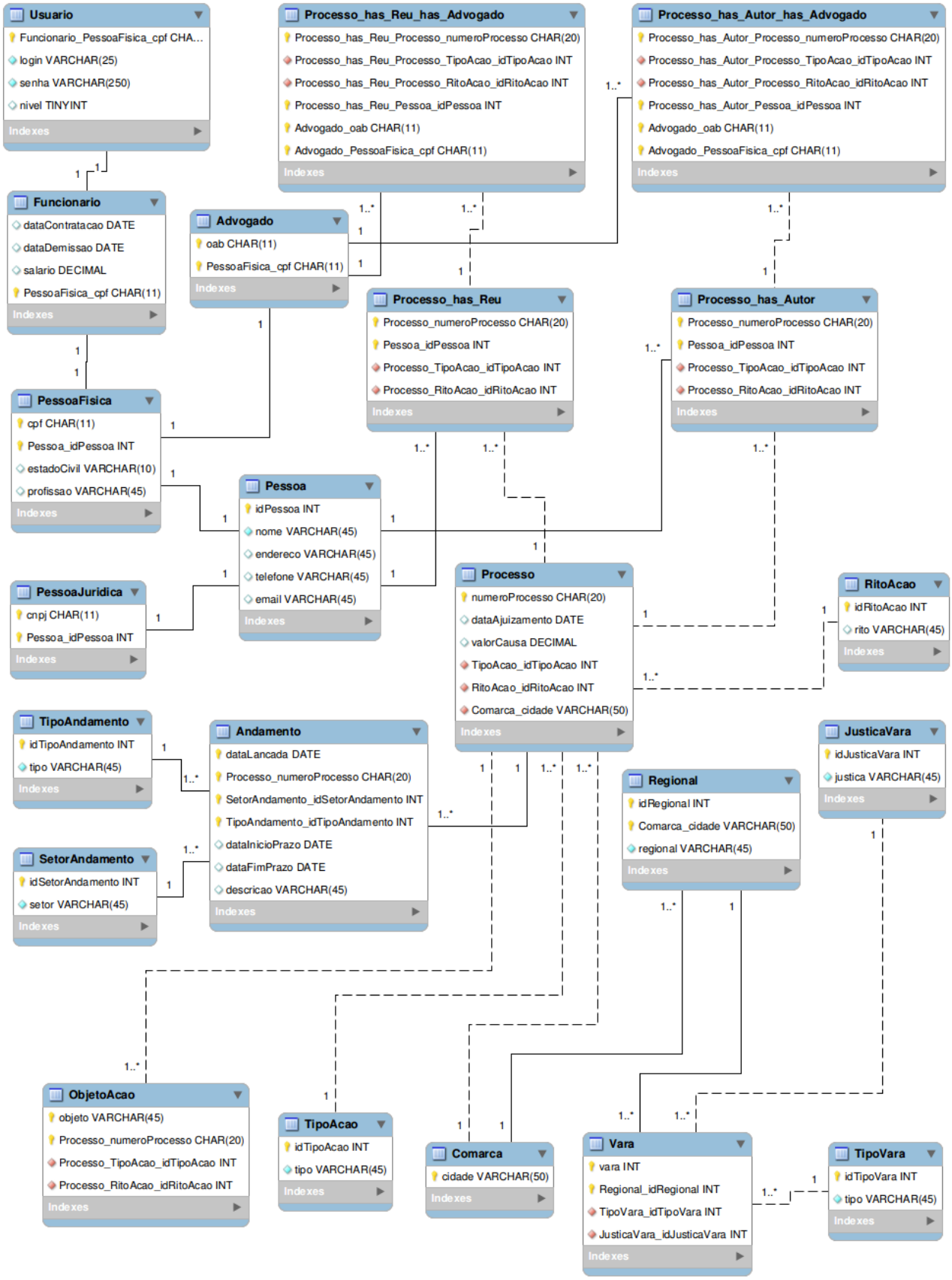
## Requisitos

1. O usuário do sistema deve poder adicionar um novo processo judicial.
2. O usuário do sistema deve poder adicionar e editar uma pessoa física e jurídica ao sistema.
3. O usuário do sistema deve poder adicionar um autor e/ou um réu a um processo.
4. O usuário do sistema deve poder adicionar um andamento a um processo.
5. O usuário deve poder deletar uma pessoa e um processo do sistema.

# Modelo simplificado



**Modelo completo original (apenas para contemplação)** – Possui alguns erros nas multiplicidades ainda não corrigidos



## Código gerado do modelo simplificado

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';


CREATE SCHEMA IF NOT EXISTS `BDSMADV_ALPHA` DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci ;
USE `BDSMADV_ALPHA` ;


-- -----------------------------------------------------
-- Table `BDSMADV_ALPHA`.`PessoaFisica`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `BDSMADV_ALPHA`.`PessoaFisica` (
  `idPessoaFisica` INT NOT NULL AUTO_INCREMENT ,
  `cpf` CHAR(25) NOT NULL ,
  `nome` VARCHAR(125) NULL ,
  `endereco` VARCHAR(125) NULL ,
  `telefone` VARCHAR(125) NULL ,
  `email` VARCHAR(125) NULL ,
  UNIQUE INDEX `cpf_UNIQUE` (`cpf` ASC) ,
  PRIMARY KEY (`idPessoaFisica`) )
ENGINE = InnoDB;



-- -----------------------------------------------------
-- Table `BDSMADV_ALPHA`.`PessoaJuridica`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `BDSMADV_ALPHA`.`PessoaJuridica` (
  `idPessoaJuridica` INT NOT NULL AUTO_INCREMENT ,
  `cnpj` CHAR(25) NOT NULL ,
  `razao` VARCHAR(125) NULL ,
  `endereco` VARCHAR(125) NULL ,
  `telefone` VARCHAR(125) NULL ,
  `email` VARCHAR(45) NULL ,
```

```
  UNIQUE INDEX `cnpj_UNIQUE` (`cnpj` ASC) ,
  PRIMARY KEY (`idPessoaJuridica`) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `BDSMADV_ALPHA`.`Processo`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `BDSMADV_ALPHA`.`Processo` (
  `idProcesso` INT NOT NULL AUTO_INCREMENT ,
  `numeroProcesso` CHAR(20) NOT NULL ,
  `dataAjuizamento` DATE NOT NULL ,
  `valorCausa` DECIMAL NOT NULL ,
  `objetoAcao` VARCHAR(125) NULL ,
  `tipoAcao` VARCHAR(125) NULL ,
  `ritoAcao` VARCHAR(125) NULL ,
  `comarca` VARCHAR(125) NULL ,
  PRIMARY KEY (`idProcesso`) ,
  UNIQUE INDEX `numeroProcesso_UNIQUE` (`numeroProcesso` ASC) )
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `BDSMADV_ALPHA`.`Andamento`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `BDSMADV_ALPHA`.`Andamento` (
  `dataLancada` DATE NOT NULL ,
  `Processo_idProcesso` INT NOT NULL ,
  `descricao` VARCHAR(250) NULL ,
  `setor` VARCHAR(125) NULL ,
  `tipo` VARCHAR(125) NULL ,
  PRIMARY KEY (`dataLancada`, `Processo_idProcesso`) ,
  INDEX `fk_Andamento_Processo1_idx` (`Processo_idProcesso` ASC) ,
  CONSTRAINT `fk_Andamento_Processo1`
    FOREIGN KEY (`Processo_idProcesso` )
    REFERENCES `BDSMADV_ALPHA`.`Processo` (`idProcesso` )
    ON DELETE CASCADE
```

```
   ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `BDSMADV_ALPHA`.`Processo_has_PessoaFisica_AUTOR`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `BDSMADV_ALPHA`.`Processo_has_PessoaFisica_AUTOR`
(
  `Processo_idProcesso` INT NOT NULL ,
  `PessoaFisica_idPessoaFisica` INT NOT NULL ,
  PRIMARY KEY (`Processo_idProcesso`, `PessoaFisica_idPessoaFisica`) ,
  INDEX `fk_Processo_has_PessoaFisica_PessoaFisica1_idx` (`PessoaFisica_idPessoaFisica` ASC)
,
  INDEX `fk_Processo_has_PessoaFisica_Processo1_idx` (`Processo_idProcesso` ASC) ,
  CONSTRAINT `fk_Processo_has_PessoaFisica_Processo1`
    FOREIGN KEY (`Processo_idProcesso` )
    REFERENCES `BDSMADV_ALPHA`.`Processo` (`idProcesso` )
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Processo_has_PessoaFisica_PessoaFisica1`
    FOREIGN KEY (`PessoaFisica_idPessoaFisica` )
    REFERENCES `BDSMADV_ALPHA`.`PessoaFisica` (`idPessoaFisica` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `BDSMADV_ALPHA`.`Processo_has_PessoaJuridica_AUTOR`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS
`BDSMADV_ALPHA`.`Processo_has_PessoaJuridica_AUTOR` (
  `Processo_idProcesso` INT NOT NULL ,
  `PessoaJuridica_idPessoaJuridica` INT NOT NULL ,
  PRIMARY KEY (`Processo_idProcesso`, `PessoaJuridica_idPessoaJuridica`) ,
  INDEX `fk_Processo_has_PessoaJuridica_PessoaJuridica1_idx`
(`PessoaJuridica_idPessoaJuridica` ASC) ,
```

```
  INDEX `fk_Processo_has_PessoaJuridica_Processo1_idx` (`Processo_idProcesso` ASC) ,
  CONSTRAINT `fk_Processo_has_PessoaJuridica_Processo1`
    FOREIGN KEY (`Processo_idProcesso` )
    REFERENCES `BDSMADV_ALPHA`.`Processo` (`idProcesso` )
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Processo_has_PessoaJuridica_PessoaJuridica1`
    FOREIGN KEY (`PessoaJuridica_idPessoaJuridica` )
    REFERENCES `BDSMADV_ALPHA`.`PessoaJuridica` (`idPessoaJuridica` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;


-- -----------------------------------------------------
-- Table `BDSMADV_ALPHA`.`Processo_has_PessoaFisica_REU`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `BDSMADV_ALPHA`.`Processo_has_PessoaFisica_REU` (
  `Processo_idProcesso` INT NOT NULL ,
  `PessoaFisica_idPessoaFisica` INT NOT NULL ,
  PRIMARY KEY (`Processo_idProcesso`, `PessoaFisica_idPessoaFisica`) ,
  INDEX `fk_Processo_has_PessoaFisica_PessoaFisica2_idx` (`PessoaFisica_idPessoaFisica` ASC)
,
  INDEX `fk_Processo_has_PessoaFisica_Processo2_idx` (`Processo_idProcesso` ASC) ,
  CONSTRAINT `fk_Processo_has_PessoaFisica_Processo2`
    FOREIGN KEY (`Processo_idProcesso` )
    REFERENCES `BDSMADV_ALPHA`.`Processo` (`idProcesso` )
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Processo_has_PessoaFisica_PessoaFisica2`
    FOREIGN KEY (`PessoaFisica_idPessoaFisica` )
    REFERENCES `BDSMADV_ALPHA`.`PessoaFisica` (`idPessoaFisica` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;
```

```
-- -----------------------------------------------------
-- Table `BDSMADV_ALPHA`.`Processo_has_PessoaJuridica_REU`
-- -----------------------------------------------------
CREATE  TABLE IF NOT EXISTS `BDSMADV_ALPHA`.`Processo_has_PessoaJuridica_REU` (
  `Processo_idProcesso` INT NOT NULL ,
  `PessoaJuridica_idPessoaJuridica` INT NOT NULL ,
  PRIMARY KEY (`Processo_idProcesso`, `PessoaJuridica_idPessoaJuridica`) ,
  INDEX `fk_Processo_has_PessoaJuridica_PessoaJuridica2_idx`
(`PessoaJuridica_idPessoaJuridica` ASC) ,
  INDEX `fk_Processo_has_PessoaJuridica_Processo2_idx` (`Processo_idProcesso` ASC) ,
  CONSTRAINT `fk_Processo_has_PessoaJuridica_Processo2`
    FOREIGN KEY (`Processo_idProcesso` )
    REFERENCES `BDSMADV_ALPHA`.`Processo` (`idProcesso` )
    ON DELETE CASCADE
    ON UPDATE CASCADE,
  CONSTRAINT `fk_Processo_has_PessoaJuridica_PessoaJuridica2`
    FOREIGN KEY (`PessoaJuridica_idPessoaJuridica` )
    REFERENCES `BDSMADV_ALPHA`.`PessoaJuridica` (`idPessoaJuridica` )
    ON DELETE CASCADE
    ON UPDATE CASCADE)
ENGINE = InnoDB;

USE `BDSMADV_ALPHA` ;



SET SQL_MODE=@OLD_SQL_MODE;
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

## Triggers

```
-- Full Trigger DDL Statements
-- Note: Only CREATE TRIGGER statements are allowed
DELIMITER $$

USE `BDSMADV_ALPHA`$$
```

```
CREATE
DEFINER=`root`@`localhost`
TRIGGER `BDSMADV_ALPHA`.`Processo_has_PessoaFisica_AUTOR_BINS`
BEFORE INSERT ON `BDSMADV_ALPHA`.`Processo_has_PessoaFisica_AUTOR`
FOR EACH ROW
-- Edit trigger body code below this line. Do not edit lines above this one

DELETE FROM Processo_has_PessoaFisica_REU
WHERE PessoaFisica_idPessoaFisica = NEW.PessoaFisica_idPessoaFisica AND
Processo_idProcesso = NEW.Processo_idProcesso$$


-- Full Trigger DDL Statements
-- Note: Only CREATE TRIGGER statements are allowed
DELIMITER $$

USE `BDSMADV_ALPHA`$$


CREATE
DEFINER=`root`@`localhost`
TRIGGER `BDSMADV_ALPHA`.`Processo_has_PessoaFisica_REU_BUPD`
BEFORE INSERT ON `BDSMADV_ALPHA`.`Processo_has_PessoaFisica_REU`
FOR EACH ROW
-- Edit trigger body code below this line. Do not edit lines above this one
DELETE FROM Processo_has_PessoaFisica_AUTOR
WHERE PessoaFisica_idPessoaFisica = NEW.PessoaFisica_idPessoaFisica AND
Processo_idProcesso = NEW.Processo_idProcesso$$


-- Full Trigger DDL Statements
-- Note: Only CREATE TRIGGER statements are allowed
DELIMITER $$

USE `BDSMADV_ALPHA`$$


CREATE
DEFINER=`root`@`localhost`
TRIGGER `BDSMADV_ALPHA`.`Processo_has_PessoaJuridica_AUTOR_BINS`
```

```
BEFORE INSERT ON `BDSMADV_ALPHA`.`Processo_has_PessoaJuridica_AUTOR`
FOR EACH ROW
-- Edit trigger body code below this line. Do not edit lines above this one

DELETE FROM Processo_has_PessoaJuridica_REU
WHERE PessoaJuridica_idPessoaJuridica = NEW.PessoaJuridica_idPessoaJuridica AND
Processo_idProcesso = NEW.Processo_idProcesso$$


-- Full Trigger DDL Statements
-- Note: Only CREATE TRIGGER statements are allowed
DELIMITER $$

USE `BDSMADV_ALPHA`$$

CREATE
DEFINER=`root`@`localhost`
TRIGGER `BDSMADV_ALPHA`.`Processo_has_PessoaJuridica_REU_BUPD`
BEFORE INSERT ON `BDSMADV_ALPHA`.`Processo_has_PessoaJuridica_REU`
FOR EACH ROW
-- Edit trigger body code below this line. Do not edit lines above this one
DELETE FROM Processo_has_PessoaJuridica_AUTOR
WHERE PessoaJuridica_idPessoaJuridica = NEW.PessoaJuridica_idPessoaJuridica AND
Processo_idProcesso = NEW.Processo_idProcesso$$
```

## Exemplo de Código de conexão com o banco

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
/**
 *
 * @author gustavo
 */
public class Conexao {

    public static Connection conectar() throws ClassNotFoundException, InstantiationException, IllegalAccessException, SQLException {
        Class.forName("com.mysql.jdbc.Driver").newInstance();
        Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/BDSMADV_ALPHA", "root", "gustavo");
        return con;
    }
}
```

## Exemplo de uso da conexão

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
```

```
import java.io.IOException;
import java.io.PrintWriter;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

/**
 *
 * @author gustavo
 */
@WebServlet(urlPatterns = {"/AdicionarPessoa"})
public class AdicionarPessoa extends HttpServlet {

    /**
     * Processes requests for both HTTP <code>GET</code> and <code>POST</code>
     * methods.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    protected void processRequest(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        response.setContentType("text/html;charset=UTF-8");
        PrintWriter out = response.getWriter();

        out.println("<!DOCTYPE html>");
        out.println("<html>");
        out.println("<head>");
        out.println("<title>Servlet AdicionarPessoa</title>");
```

```
        out.println("</head>");
        out.println("<body>");

        Connection con;
        PreparedStatement stmt;
        String query;

        try {
            con = Conexao.conectar();

            String nome = request.getParameter("nome");
            String endereco = request.getParameter("endereco");
            String telefone = request.getParameter("telefone");
            String email = request.getParameter("email");
            String cpf_cnpj = request.getParameter("cpf_cnpj");
            String tipo = request.getParameter("tipo");

            if (tipo.equals("fisica")) {
                query = "INSERT INTO PessoaFisica (nome, endereco, telefone, email, cpf) "
                        + "VALUES('" + nome + "', '" + endereco + "', '" + telefone + "', '" + email + "', '" +
cpf_cnpj + "');";
            }
            else {
                query = "INSERT INTO PessoaJuridica (razao, endereco, telefone, email, cnpj) "
                        + "VALUES('" + nome + "', '" + endereco + "', '" + telefone + "', '" + email + "', '" +
cpf_cnpj + "');";
            }
            stmt = con.prepareStatement(query);
            stmt.executeUpdate(query);
            response.sendRedirect("/GSMA/ListarPessoas");
            con.close();
        } catch (SQLException ex) {
            out.println("Erro SQL<br />" + ex.getMessage());
        } catch (ClassNotFoundException ex) {
            out.println("Erro Class");
        } catch (InstantiationException ex) {
            out.println("Erro Instanciação");
```

```
        } catch (IllegalAccessException ex) {
            out.println("Erro de Acesso Ilegal");
        } catch (Exception e) {
            out.println(e.getMessage());
        }
        out.println("</body>");
        out.println("</html>");
        out.close();
    }

    // <editor-fold defaultstate="collapsed" desc="HttpServlet methods. Click on the + sign on the left to edit the code.">
    /**
     * Handles the HTTP <code>GET</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        processRequest(request, response);
    }

    /**
     * Handles the HTTP <code>POST</code> method.
     *
     * @param request servlet request
     * @param response servlet response
     * @throws ServletException if a servlet-specific error occurs
     * @throws IOException if an I/O error occurs
     */
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
```

```
        processRequest(request, response);
    }

    /**
     * Returns a short description of the servlet.
     *
     * @return a String containing servlet description
     */
    @Override
    public String getServletInfo() {
        return "Short description";
    }// </editor-fold>

}
```