

Universidade Federal do Rio Grande do Norte

Instituto Metr pole Digital

Banco de Dados

Manoel Messias De Barros Junior

Especifica  o do Projeto de Banco de Dados

Sistema de Gerenciamento de Chamados de TI

Natal

Introdução:

Este Projeto foi elaborado como trabalho de conclusão da disciplina Banco de Dados do Instituto Metrópole Digital. Trata-se de uma aplicação destinado a um sistema de gestão de chamados e incidentes de Ti para uma empresa.

O Projeto é baseado nas melhores práticas do framework de gestão de serviços de TI ITIL V3. Para tanto implementa a gestão de nível de serviço (SLA - Service Level Agreement), Catálogo de Serviços, Inserção de ações nos chamados que permitem agendamento, Formas de solicitação de um chamado e pesquisa de satisfação com o usuário que registrou o chamado.

A gestão de chamados e incidentes que utiliza o modelo do ITIL V3 permite uma direção estratégica dos serviços de TI, beneficiando a organização através de melhoria continuada, acompanhando os incidentes por todo o seu ciclo de vida.

Objetivos do Sistema:

Nesse sistema, usuários e analistas de TI podem abrir chamados, mas apenas os analistas possuem acesso à interface para encerra-los.

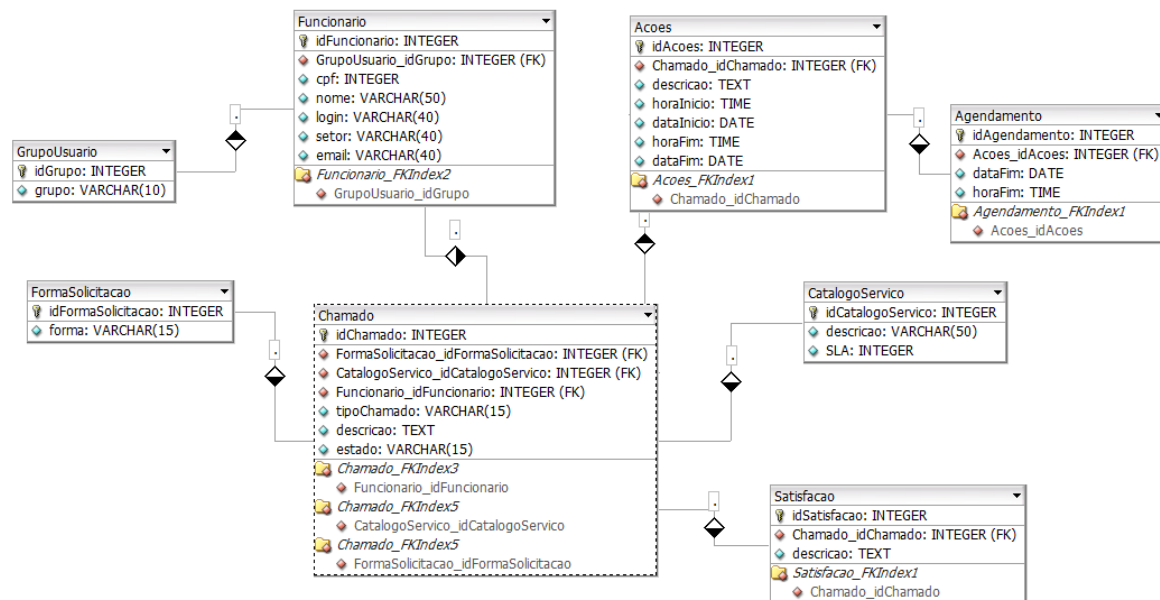
Os chamados possuem Descrição, Tipo [Incidente, Problema, Solicitação, Erro Conhecido], Forma de Solicitação [Telefone, Email, Frontend, Pessoalmente], estado (se está aberto ou fechado). Os chamados estão associados a um catálogo de serviços que definem supergrupos genéricos de incidentes onde cada tupla tem um tempo de SLA associado.

Cada chamado possui uma ou mais ações, que indicam o que foi executado pela(s) equipe(s). Se um chamado não possui ação, ele ainda não está em atendimento. Cada tupla das ações possui agendamento ou não, quando o analista informa qual o horário poderá executar o atendimento. Além disso as ações possuem descrição (onde o analista detalha o que foi executado nessa ação), hora/data de inicio e hora/data de fim, que é apontará o tempo despendido para executar determinada ação.

Ao encerrar o chamado, o usuário receberá uma pesquisa de satisfação. Todos os chamados encerrados têm uma pesquisa de satisfação associada.

As regras de negócio da aplicação foram todas implementadas no código SQL. Essas regras estão detalhadas através de comentários dos códigos, verificar a seção **Script de Criação dos Elementos do BD**.

Diagrama DER



Script de Criação dos Elementos do BD

```
--Cria o Banco de dados Projeto Final
```

```
CREATE DATABASE "ProjetoFinal"
```

```
WITH
```

```
OWNER = postgres
```

```
ENCODING = 'UTF8'
```

```
LC_COLLATE = 'Portuguese_Brazil.1252'
```

```
LC_CTYPE = 'Portuguese_Brazil.1252'
```

```
TABLESPACE = pg_default
```

```
CONNECTION LIMIT = -1;
```

```
-- *****
```

```
-- Criação das tabelas *
```

```
-- *****
```

```
-- Cria a tabela catálogo de Serviços: Tipos genéricos de Problemas.
```

```

-- CRIAR REGRA: PERMITIR INSERÇÃO NESSA TABELA APENAS ADMIN
CREATE TABLE CatalogoServicos (
    idCatalogoServicos SERIAL NOT NULL UNIQUE,
    descricao VARCHAR(50) NOT NULL,
    SLA INTEGER NOT NULL,
    PRIMARY KEY(idCatalogoServicos));

-- Cria a tabela Forma de Solicitação.
-- Telefone, email, frontend, pessoalmente (APENAS)
CREATE TABLE FormaSolicitacao (
    idFormaSolicitacao SERIAL NOT NULL UNIQUE,
    forma VARCHAR(15) NOT NULL,
    PRIMARY KEY(idFormaSolicitacao));

-- -----
-- User
-- or
-- Admin
-- -----

-- O usuário pode ser um usuário comum OU Admin (Analista de TI)
CREATE TABLE GrupoUsuario (
    idGrupo SERIAL NOT NULL UNIQUE,
    grupo VARCHAR(10) NOT NULL,
    PRIMARY KEY(idGrupo));

-- O usuário pode ser um usuário comum OU Admin (Analista de TI)
COMMENT ON TABLE GrupoUsuario IS 'O usuário pode ser um usuário comum ou Admin
(Analista de TI) - User or Admin';

/*
Cria a tabela de Funcionário. Nas Nossas regras de negócio, não importa o cargo
do colaborador
há uma chave estrangeira para o grupo do colaborador, que pode ser ADMIN, Comum
*/

CREATE TABLE Funcionarios (
    idFuncionario SERIAL NOT NULL UNIQUE,
    GrupoUsuario_idGrupo INTEGER NULL,
    cpf VARCHAR(14) UNIQUE,
    nome VARCHAR(50),
    login VARCHAR(40),
    setor VARCHAR(40),
    email VARCHAR(40),
    PRIMARY KEY(idFuncionario),

```

```
FOREIGN KEY(GrupoUsuario_idGrupo) REFERENCES GrupoUsuario(idGrupo));
```

-- Um chamado possui uma forma de solicitação, um Serviço genérico, um funcionário atrelado e ações

```
CREATE TABLE Chamados (  
    idChamado SERIAL NOT NULL UNIQUE,  
    FormaSolicitacao_idFormaSolicitacao INTEGER NOT NULL,  
    CatalogoServico_idCatalogoServicos INTEGER NOT NULL ,  
    Funcionario_idFuncionario INTEGER NOT NULL,  
    tipoChamado VARCHAR(15) check(  
        tipoChamado = 'Incidente' or  
        tipoChamado = 'Problema' or  
        tipoChamado = 'Solicitação'  
    ),  
    descricao TEXT,  
    estado VARCHAR(15),  
    PRIMARY KEY(idChamado),  
    FOREIGN KEY(Funcionario_idFuncionario)  
        REFERENCES Funcionarios(idFuncionario),  
    FOREIGN KEY(CatalogoServico_idCatalogoServicos)  
        REFERENCES CatalogoServicos(idCatalogoServicos),  
    FOREIGN KEY(FormaSolicitacao_idFormaSolicitacao)  
        REFERENCES FormaSolicitacao(idFormaSolicitacao));
```

-- Cria a tabela Ações. Uma ação trata-se de um dos apontamentos de um chamado. Uma ação pode ou não ter um agendamento.

```
CREATE TABLE Acoes (  
    idAcoes SERIAL NOT NULL UNIQUE,  
    Chamado_idChamado INTEGER NOT NULL ,  
    descricao TEXT,  
    horaInicio TIME,  
    dataInicio DATE,  
    horaFim TIME,  
    dataFim DATE,  
    PRIMARY KEY(idAcoes),  
    FOREIGN KEY(Chamado_idChamado)  
        REFERENCES Chamados(idChamado));
```

-- CRIA TABELA AGENDAMENTO. Nem toda AÇÃO tem agendamento, MAS TODO AGENDAMENTO TEM UMA AÇÃO

```
CREATE TABLE Agendamentos (  
    idAgendamento SERIAL NOT NULL UNIQUE,  
    Acoes_idAcoes INTEGER NOT NULL,  
    dataFim DATE,  
    horaFim TIME,
```

```

PRIMARY KEY(idAgendamento),
  FOREIGN KEY(Acoes_idAcoes)
    REFERENCES Acoes(idAcoes));

/*
Cria Tabela Pesquisa de Satisfacao.
Toda pesquisa de satisfação está associada a um chamado.
A aplicação só insere a pesquisa de satisfação após o encerramento de um chamado.
Enquanto o chamado estiver aberto, não haverá pesquisa preenchida.
*/

CREATE TABLE Satisfacao (
  idSatisfacao SERIAL NOT NULL UNIQUE,
  Chamado_idChamado INTEGER NOT NULL,
  descricao TEXT,
PRIMARY KEY(idSatisfacao) ,
  FOREIGN KEY(Chamado_idChamado)
    REFERENCES Chamados(idChamado));

-- *****
-- FIM -- Criação das tabelas -- FIM  *
-- *****

-- VIEW 01 -- Todos Colaboradores do RH
CREATE VIEW view_funcionarios_RH as
SELECT * FROM funcionarios f WHERE f.setor='Recursos Humanos';
-----

-- VIEW 02 -- Visualizar todos os chamados em aberto
CREATE VIEW view_chamados_abertos as
SELECT * FROM Chamados c WHERE c.estado='Aberto';
-----

-- VIEW 03 -- VISUALIZAR CHAMADOS ENCERRADOS
CREATE VIEW view_chamados_fechados as
SELECT * FROM Chamados c WHERE c.estado='Fechado';
-----

-- VIEW 04 - TABELA DE CHAMADOS ABERTOS FORMATADA
CREATE VIEW view_abertos_formato as
SELECT chamados.idchamado, formasolicitacao.forma, chamados.tipochamado,
chamados.descricao, funcionarios.nome
FROM chamados,funcionarios,formasolicitacao, catalogoservicos
WHERE chamados.estado='Aberto'
AND chamados.funcionario_idfuncionario=funcionarios.idfuncionario
AND
chamados.formasolicitacao_idformasolicitacao=formasolicitacao.idformasolicitacao

```

```

AND catalogoservicos.idcatalogoservicos =
chamados.catalogoservico_idcatalogoservicos;
-----

-- VIEW 05 - TABELA DE CHAMADOS ABERTOS FORMATADA (Faz consulta semelhante
anterior, mas utilizando INNER JOIN)
CREATE VIEW view_fechados_formato as
SELECT chamados.idchamado, formasolicitacao.forma, chamados.tipochamado,
chamados.descricao, funcionarios.nome
FROM chamados
INNER JOIN funcionarios ON
funcionarios.idfuncionario=chamados.funcionario_idfuncionario
INNER JOIN formasolicitacao ON
chamados.formasolicitacao_idformasolicitacao=formasolicitacao.idformasolicitacao
INNER JOIN catalogoservicos ON catalogoservicos.idcatalogoservicos =
chamados.catalogoservico_idcatalogoservicos
WHERE chamados.estado='Fechado';
-----

-- VIEW 06 - VISUALIZAR PESQUISAS DE SATISFACAO
-- Exibe Problema do chamado e a pesquisa de satisfacao para chamados encerrados
-----

CREATE VIEW view_satisfacao as
SELECT chamados.idchamado, funcionarios.nome, Chamados.descricao,
Satisfacao.descricao
FROM chamados
INNER JOIN funcionarios ON
funcionarios.idfuncionario=chamados.funcionario_idfuncionario
INNER JOIN Satisfacao on satisfacao.Chamado_idChamado=chamados.idchamado
WHERE chamados.estado='Fechado';

-----
-----

-- FUNÇÃO/TRIGGER VERIFICAR PADRÃO DE EXPRESSAO REGULAR PARA CPF, pattern
123.456.789-10 --
-----
-----

CREATE FUNCTION check_CPF_Regex() RETURNS Trigger AS $check_CPF_Regex$
BEGIN
    IF NEW.cpf NOT SIMILAR TO '[0-9]{3}\.[0-9]{3}\.[0-9]{3}\-[0-9]{2}' THEN
        RAISE EXCEPTION 'PADRAO REGEX INCORRETO PARA O CPF';
    END IF;
RETURN NEW;
END;
$check_CPF_Regex$ LANGUAGE plpgsql;

CREATE Trigger cpf_insert BEFORE INSERT OR UPDATE

```

```

ON Funcionarios
FOR EACH ROW EXECUTE
PROCEDURE check_CPF_Regex();
-----
-----
-- VERIFICAR PADRÃO DE EXPRESSÃO REGULAR PARA EMAIL, pattern
nome.sobrenome@empresa.com.br --
-----
-----
CREATE FUNCTION check_email_Regex() RETURNS Trigger AS $check_email_Regex$
BEGIN
    IF NEW.email NOT SIMILAR TO '%\.%\@empresa.com.br' THEN
        RAISE EXCEPTION 'PADRAO REGEX INCORRETO PARA O EMAIL';
    END IF;
RETURN NEW;
END;
$check_email_Regex$ LANGUAGE plpgsql;

CREATE Trigger email_insert BEFORE INSERT OR UPDATE
ON Funcionarios
FOR EACH ROW EXECUTE
PROCEDURE check_email_Regex();
-----
-----
-- VERIFICAR PADRAO DE EXPRESSÃO REGULAR PARA LOGIN, pattern nome.sobrenome --
-----
CREATE FUNCTION check_login_Regex() RETURNS Trigger AS $check_login_Regex$
BEGIN
    IF NEW.login NOT SIMILAR TO '%\.%' THEN
        RAISE EXCEPTION 'PADRAO REGEX INCORRETO PARA O LOGIN';
    END IF;
RETURN NEW;
END;
$check_login_Regex$ LANGUAGE plpgsql;

CREATE Trigger login_insert BEFORE INSERT OR UPDATE
ON Funcionarios
FOR EACH ROW EXECUTE
PROCEDURE check_login_Regex();
-----
-----
-- FUNÇÃO EM Pl/pgSQL PARA ENCERRAR CHAMADO --
-- Utilizado na aplicação como "SELECT encerrar_chamado();"
-----
CREATE FUNCTION encerrar_chamado(chamado INTEGER) RETURNS void AS $$
BEGIN

```



```

UPDATE Chamados
SET estado = 'Fechado'
WHERE chamados.idChamado=chamado;
END;
$$ LANGUAGE plpgsql;

```

```

/*****
FUNÇÃO EM Pl/pgSQL para Cadastrar novo funcionário
*****/
CREATE FUNCTION cadastrar_funcionario_comum(_cpf VARCHAR(14),_nome VARCHAR(50),
_login VARCHAR(40), _setor VARCHAR(40), _email VARCHAR(40))
RETURNS void AS $$
DECLARE
    grupo INTEGER := 2;
BEGIN
    INSERT INTO Funcionarios(grupo,cpf,nome, login,setor, email) VALUES (grupo,
_cpf, _nome, _login, _setor, _email);
END;
$$ LANGUAGE plpgsql;

```

```

/*****
NÃO AUTORIZAR DELETE na tabela AÇÕES
*****/
CREATE FUNCTION undelete_acoes() RETURNS Trigger AS $undelete_acoes$
BEGIN
    RAISE EXCEPTION 'NÃO É PERMITIDO APAGAR ACOES';
END;
$undelete_acoes$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER delete_acao BEFORE DELETE
ON Acoes
FOR EACH ROW EXECUTE
PROCEDURE undelete_acoes();

```

```

/*****
NÃO AUTORIZAR DELETE na tabela CHAMADOS
*****/
CREATE FUNCTION undelete_chamados() RETURNS Trigger AS $undelete_chamados$
BEGIN
    RAISE EXCEPTION 'NÃO É PERMITIDO APAGAR CHAMADOS.';
END;
$undelete_chamados$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER delete_chamado BEFORE DELETE
ON Chamados

```

```
FOR EACH ROW EXECUTE  
PROCEDURE undelete_chamados();
```

O código abaixo Popula as tabelas para que possamos realizar consultas e trabalhar com as funções apresentadas.

```
/*  
SCRIPT PARA POPULAR O BANCO CRIADO  
*/  
  
INSERT INTO GrupoUsuario(grupo) values('Admin');  
INSERT INTO GrupoUsuario(grupo) values('User');  
  
INSERT INTO FormaSolicitacao(forma) values('Telefone');  
INSERT INTO FormaSolicitacao(forma) values('Email');  
INSERT INTO FormaSolicitacao(forma) values('Frontend');  
INSERT INTO FormaSolicitacao(forma) values('Pessoalmente');  
  
INSERT INTO CatalogoServicos(descricao,SLA) values('Assuntos  
Administrativos',10080);  
-- SLA em minutos, 7 dias  
INSERT INTO CatalogoServicos(descricao,SLA) values('Automação de TI',240);  
-- SLA 4 HORAS  
INSERT INTO CatalogoServicos(descricao,SLA) values('Dispositivos Móveis',120);  
-- SLA 2 HORAS  
INSERT INTO CatalogoServicos(descricao,SLA) values('Gestão de Acesso',1440);  
-- SLA 1 DIA  
INSERT INTO CatalogoServicos(descricao,SLA) values('Infraestrutura',60);  
-- SLA 1 HORA  
INSERT INTO CatalogoServicos(descricao,SLA) values('Datacenter',90);  
-- SLA 1 HORA E 30 MIN  
INSERT INTO CatalogoServicos(descricao,SLA) values('Pacote MS Office',180);  
-- SLA 3 HORAS  
INSERT INTO CatalogoServicos(descricao,SLA) values('Rede-Conectividade',45);  
-- SLA 45 MINUTOS  
INSERT INTO CatalogoServicos(descricao,SLA) values('E-MAIL',150);  
-- SLA 2 HORAS E 30 MIN  
INSERT INTO CatalogoServicos(descricao,SLA) values('Segurança da Informação',80);  
-- SLA 1 HORA E 20 MIN  
INSERT INTO CatalogoServicos(descricao,SLA) values('Dúvida-Consultoria',300);  
-- SLA 5 horas  
INSERT INTO CatalogoServicos(descricao,SLA) values('Softwares',330);  
-- SLA 5 HORAS E 30 MIN  
INSERT INTO CatalogoServicos(descricao,SLA) values('Sistemas',210);
```

-- SLA 3 HORAS E 30 MIN

INSERT INTO CatalogoServicos(descricao,SLA) values('Telefonia',2880);

-- SLA 2 DIAS

Begin Transaction;

INSERT INTO Funcionarios(GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (1,'074.320.414-07','Manoel Barros','Manoel.Barros','Tecnologia da
Informação','manoel.barros@empresa.com.br');

INSERT INTO Funcionarios(GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (1,'690.586.070-03','Joao Silva','Joao.Silva','Tecnologia da
Informação','joao.silva@empresa.com.br');

INSERT INTO Funcionarios(GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (1,'661.002.164-34','Alex Andrade','Alex.Andrade','Tecnologia da
Informação','Alex.Andrade@empresa.com.br');

INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'901.314.110-20','Maria
Siqueira','maria.siqueira','Controladoria','maria.siqueira@empresa.com.br');

INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'725.633.070-74','Antonio
Andrade','antonio.andrade','Controladoria','antonio.andrade@empresa.com.br');

INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'428.525.960-59','Jaqueline
Santos','jaqueline.santos','Controladoria','jaqueline.santos@empresa.com.br');

INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'823.137.610-02','Alberto
Batista','alberto.batista','Almoxarifado','alberto.batista@empresa.com.br');

INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'469.388.360-35','Maria Roberta
Junqueira','maria.junqueira','Almoxarifado','maria.junqueira@empresa.com.br');

INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'566.683.220-42','Hudson Macedo','hudson.macedo','Recursos
Humanos','hudson.macedo@empresa.com.br');

INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'892.659.380-05','Erick Bento Campos','erick.campos','Recursos
Humanos','erick.campos@empresa.com.br');

INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'020.735.350-62','Thales Raul Manoel
Caldeira','tales.caldeira','Recursos Humanos','tales.caldeira@empresa.com.br');

INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'645.895.790-49','Eduarda Bianca Stefany
Galvão','eduarda.galvao','Financeiro','eduarda.galvao@empresa.com.br');

INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'384.780.150-35','Milena Tereza Fátima
Aragão','milena.aragao','Comercial','milena.aragao@empresa.com.br');

```

INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'060.599.740-31','Clarice Sara Luiza
Brito','clarice.brito','Financeiro','clarice.brito@empresa.com.br');
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'912.029.130-24','Pietro Tiago Diogo
Fernandes','pietro.fernandes','Almoxarifado','pietro.fernandes@empresa.com.br');
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'686.439.190-63','Tiago César
Fogaça','tiago.fogaca','Estoque','tiago.fogaca@empresa.com.br');
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'568.805.520-05','Jéssica Luiza
Viana','jessica.viana','Manutenção','jessica.viana@empresa.com.br');
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'388.628.550-28','Levi Bernardo Vicente
Rezende','levi.rezende','Manutenção','levi.rezende@empresa.com.br');
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'234.844.930-70','Juliana Betina
Cavalcanti','juliana.cavalcanti','Comercial','juliana.cavalcanti@empresa.com.br')
;
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'752.017.800-59','Nicole Manuela Camila
Araújo','nicole.araujo','Recepção','nicole.araujo@empresa.com.br');
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'446.027.590-25','Luiza
Daiane','luiza.daiane','Recepção','luiza.daiane@empresa.com.br');
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'914.218.260-38','Daniel Benedito dos
Santos','daniel.santos','Segurança do Trabalho','daniel.santos@empresa.com.br');
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'297.240.740-72','Renata Stefany Analu
Freitas','renata.freitas','Ambulatório','renata.freitas@empresa.com.br');
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'955.118.430-08','Brenda Luciana Daniela da
Paz','brenda.paz','Segurança do Trabalho','brenda.paz@empresa.com.br');
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'151.618.940-08','Iago Renan Nelson
Sales','iago.sales','Ambulatório','iago.sales@empresa.com.br');
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'442.027.520-07','Rafael Lorenzo
Dias','rafael.dias','Ambulatório','rafael.dias@empresa.com.br');
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'705.055.590-80','Bruno Sebastião Cardoso','bruno.cardoso','Segurança
do Trabalho','bruno.cardoso@empresa.com.br');

```

```
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'022.942.990-43','Patrícia Rosa
Batista','patricia.batista','Compras','patricia.batista@empresa.com.br');
```

```
INSERT INTO Funcionarios (GrupoUsuario_idGrupo,cpf,nome, login,setor, email)
VALUES (2,'741.099.160-70','Leonardo Isaac Gustavo da
Rocha','leandro.rocha','Comercial','leandro.rocha@empresa.com.br');
Commit Transaction;
```

```
-- TIPOChamados: INCIDENTE, PROBLEMA, SOLICITAÇÃO
```

```
INSERT INTO Chamados
(FormaSolicitacao_idFormaSolicitacao,CatalogoServico_idCatalogoServicos,Funcionar
io_idFuncionario,tipoChamado,descricao,estado)
VALUES (1,5,12,'Incidente','Por gentileza, substituir meu mouse que está dando
clique duplo','Aberto');
```

```
INSERT INTO Chamados
(FormaSolicitacao_idFormaSolicitacao,CatalogoServico_idCatalogoServicos,Funcionar
io_idFuncionario,tipoChamado,descricao,estado)
VALUES (2,4,20,'Incidente','Não consigo logar na máquina','Aberto');
```

```
INSERT INTO Chamados
(FormaSolicitacao_idFormaSolicitacao,CatalogoServico_idCatalogoServicos,Funcionar
io_idFuncionario,tipoChamado,descricao,estado)
VALUES (2,14,19,'Problema','Não consigo acessar o facebook','Aberto');
```

```
INSERT INTO Chamados
(FormaSolicitacao_idFormaSolicitacao,CatalogoServico_idCatalogoServicos,Funcionar
io_idFuncionario,tipoChamado,descricao,estado)
VALUES (4,4,11,'Solicitação','Atualizar o Office para versão 365','Aberto');
```

```
INSERT INTO Chamados
(FormaSolicitacao_idFormaSolicitacao,CatalogoServico_idCatalogoServicos,Funcionar
io_idFuncionario,tipoChamado,descricao,estado)
VALUES (3,3,7,'Incidente','Celular da empresa não faz lgação','Aberto');
```

```
INSERT INTO Chamados
(FormaSolicitacao_idFormaSolicitacao,CatalogoServico_idCatalogoServicos,Funcionar
io_idFuncionario,tipoChamado,descricao,estado)
VALUES (3,13,8,'Problema','ERP não abre no meu perfil','Fechado');
```

```
INSERT INTO Chamados
(FormaSolicitacao_idFormaSolicitacao,CatalogoServico_idCatalogoServicos,Funcionar
io_idFuncionario,tipoChamado,descricao,estado)
VALUES (4,10,9,'Solicitação','Verificar email recebido - phishing','Fechado');
```

```

INSERT INTO Chamados
(FormaSolicitacao_idFormaSolicitacao,CatalogoServico_idCatalogoServicos,Funcionario_idFuncionario,tipoChamado,descricao,estado)
VALUES (3,13,15,'Problema','Aparelho Telefonico está mudo.','Fechado');
-----
----- no insert final, corrigir o valor Chamado_idChamado de 12 para 6
INSERT INTO Satisfacao (Chamado_idChamado,descricao) VALUES (6,'Fui muito bem
atendido, equipe solucionou rapidamente.');
```

----- no insert final, corrigir o valor Chamado_idChamado de 13 para 7

```

INSERT INTO Satisfacao (Chamado_idChamado,descricao) VALUES (7,'Equipe encerrou o
chamado sem atender o incidente.');
```

----- no insert final, corrigir o valor Chamado_idChamado de 14 para 8

```

INSERT INTO Satisfacao (Chamado_idChamado,descricao) VALUES (8,'Substituíram por
um telefone novo, com mais funções que o anterior. Nota 10');
```

```

/*
idAcoes SERIAL NOT NULL UNIQUE,
    Chamado_idChamado INTEGER NOT NULL ,
    descricao TEXT,
    horaInicio TIME,
    dataInicio DATE,
    horaFim TIME,
    dataFim DATE,
*/
INSERT INTO acoes
(Chamado_idChamado,descricao,horaInicio,dataInicio,horaFim,dataFim) VALUES
(1,'Usuário estava ausente','15:30','05-11-2018','15:50','05-11-2018');
INSERT INTO acoes
(Chamado_idChamado,descricao,horaInicio,dataInicio,horaFim,dataFim) VALUES
(1,'Efetuada substituição do mouse','07:30','06-11-2018','07:50','06-11-2018');
INSERT INTO acoes
(Chamado_idChamado,descricao,horaInicio,dataInicio,horaFim,dataFim) VALUES
(2,'Verificado ponto de rede, falha não identificada','09:00','04-11-
2018','09:15','04-11-2018');
INSERT INTO acoes
(Chamado_idChamado,descricao,horaInicio,dataInicio,horaFim,dataFim) VALUES
(2,'Identificado cabo rompido, será substituído','12:20','04-11-
2018','12:30','04-11-2018');
INSERT INTO acoes
(Chamado_idChamado,descricao,horaInicio,dataInicio,horaFim,dataFim) VALUES
(3,'Usuário foi orientado sobre o bloqueio de Redes sociais ','09:34','05-11-
2018','09:39','05-11-2018');
INSERT INTO acoes
(Chamado_idChamado,descricao,horaInicio,dataInicio,horaFim,dataFim) VALUES
```

```

(4,'Realizada tentativa de atualização sem sucesso','10:23','05-11-
2018','11:07','05-11-2018');
INSERT INTO acoes
(Chamado_idChamado,descricao,horaInicio,dataInicio,horaFim,dataFim) VALUES
(5,'Identificamos que colaborador danificou o equipamento','09:00','06-11-
2018','09:15','06-11-2018');
INSERT INTO acoes
(Chamado_idChamado,descricao,horaInicio,dataInicio,horaFim,dataFim) VALUES
(5,'Realizada substituição do aparelho','10:15','06-11-2018','10:45','06-11-
2018');
INSERT INTO acoes
(Chamado_idChamado,descricao,horaInicio,dataInicio,horaFim,dataFim) VALUES
(6,'Reinstalada aplicação, funcionou corretamente','09:25','06-11-
2018','09:45','06-11-2018');
INSERT INTO acoes
(Chamado_idChamado,descricao,horaInicio,dataInicio,horaFim,dataFim) VALUES
(7,'Identificado que colaborador usa o email da empresa para se cadastrar em
promoções','13:28','06-11-2018','13:32','06-11-2018');
INSERT INTO acoes
(Chamado_idChamado,descricao,horaInicio,dataInicio,horaFim,dataFim) VALUES
(7,'Orientamos o colaborador sobre a boa postura de utilizacao dos recursos da
empresa','16:16','06-11-2018','16:42','06-11-2018');
INSERT INTO acoes
(Chamado_idChamado,descricao,horaInicio,dataInicio,horaFim,dataFim) VALUES
(7,'Colaborador Insistiu em descumprir as boas práticas dos recursos de
informatica, chamado encerrado','07:21','07-11-2018','07:25','06-11-2018');
INSERT INTO acoes
(Chamado_idChamado,descricao,horaInicio,dataInicio,horaFim,dataFim) VALUES
(8,'Identificado defeito no aparelho de ramal, será substituído','14:28','06-11-
2018','15:13','06-11-2018');
INSERT INTO acoes
(Chamado_idChamado,descricao,horaInicio,dataInicio,horaFim,dataFim) VALUES
(8,'Aparelho foi substituído','08:56','07-11-2018','09:23','07-11-2018');

/*
CREATE TABLE Agendamentos (
    idAgendamento SERIAL NOT NULL UNIQUE,
    Acoes_idAcoes INTEGER NOT NULL,
    dataFim DATE,
    horaFim TIME,
PRIMARY KEY(idAgendamento),
FOREIGN KEY(Acoes_idAcoes)
REFERENCES Acoes(idAcoes));
*/

```

```
INSERT INTO Agendamentos (Acoes_idAcoes,dataFim,horaFim) values (1,'05-11-2018','09:00');  
INSERT INTO Agendamentos (Acoes_idAcoes,dataFim,horaFim) values (3,'04-11-2018','12:00');  
INSERT INTO Agendamentos (Acoes_idAcoes,dataFim,horaFim) values (11,'05-11-2018','07:00');
```

Conclusão

Senti algumas dificuldades para conclusão desse projeto. Uma delas foi de que, após minhas buscas, o PostgreSQL não permite inserir controle de transações dentro de funções definidas pela linguagem PL/PgSQL. Com a Elaboração desse projeto pude colocar em prática maior parte do conhecimento adquirido na disciplina de Banco de Dados, fato que se mostrou imensamente importante para a minha formação.