**Criação das Tabelas**

**Tabela de Auditoria**

CREATE TABLE GO\_AUDITORIA

(

ID NUMBER (16,2) NOT NULL ,

TABELA VARCHAR2 (30) NOT NULL ,

TIPO\_ALTERACAO VARCHAR2 (10) NOT NULL ,

HISTORICO VARCHAR2 (4000) ,

DATA DATE NOT NULL ,

USUARIO VARCHAR2 (300) NOT NULL

) ;

ALTER TABLE GO\_AUDITORIA ADD CHECK ( TIPO\_ALTERACAO IN ('DELETE', 'UPDATE')) ;

COMMENT ON COLUMN GO\_AUDITORIA.ID IS 'Mensagem que o usuario esta enviando para o GOLPHERS' ;

COMMENT ON COLUMN GO\_AUDITORIA.TABELA IS 'Contem o nome da tabela que recebeu a alteracao' ;

COMMENT ON COLUMN GO\_AUDITORIA.TIPO\_ALTERACAO IS 'Informar se foi UPDATE ou DELETE' ;

COMMENT ON COLUMN GO\_AUDITORIA.HISTORICO IS 'CASO SEJA ALTERAÇÃO, INFORMAR OS CAMPOS QUE FORAM ALTERADOS' ;

COMMENT ON COLUMN GO\_AUDITORIA.DATA IS 'Data que foi realizada a alteração ou exclusão' ;

COMMENT ON COLUMN GO\_AUDITORIA.USUARIO IS 'LOGIN DO USUARIO QUE REALIZOU A ALTERAÇÃO' ;

ALTER TABLE GO\_AUDITORIA ADD CONSTRAINT GO\_AUDITORIA\_PK PRIMARY KEY ( ID ) ;

**Tabela de Categoria**

CREATE TABLE GO\_CATEGORIA

(

ID NUMBER (16,2) NOT NULL ,

CATEGORIA VARCHAR2 (200) NOT NULL ,

STA\_ATIVO VARCHAR2 (1) DEFAULT 'N' NOT NULL

) ;

ALTER TABLE GO\_CATEGORIA ADD CONSTRAINT GOCATE\_STAT\_CK CHECK ( STA\_ATIVO IN ('N', 'S')) ;

COMMENT ON TABLE GO\_CATEGORIA IS 'Tabela de Categoria' ;

COMMENT ON COLUMN GO\_CATEGORIA.ID IS 'identificador da tabela de categoria' ;

COMMENT ON COLUMN GO\_CATEGORIA.CATEGORIA IS 'informa a descricao da categoria' ;

COMMENT ON COLUMN GO\_CATEGORIA.STA\_ATIVO IS 'Mostra se a categoria e as noticias associadas a ela, serao ou nao exibidas' ;

ALTER TABLE GO\_CATEGORIA ADD CONSTRAINT GO\_CATEGORIA\_PK PRIMARY KEY ( ID ) ;

ALTER TABLE GO\_CATEGORIA ADD CONSTRAINT CATE\_STA\_ATIVO\_UN UNIQUE ( CATEGORIA , STA\_ATIVO ) ;

**Tabela de Contato**

CREATE TABLE GO\_CONTATO

(

ID NUMBER (16,2) NOT NULL ,

DT\_MENSAGEM DATE NOT NULL ,

MENSAGEM VARCHAR2 (2000) DEFAULT sysdate NOT NULL ,

GO\_PESSOA\_ID NUMBER (16,2) NOT NULL

) ;

COMMENT ON COLUMN GO\_CONTATO.ID IS 'identificador da tabela de contato' ;

COMMENT ON COLUMN GO\_CONTATO.DT\_MENSAGEM IS 'data que a mensagem foi postada - SYSDATE' ;

COMMENT ON COLUMN GO\_CONTATO.MENSAGEM IS 'Mensagem que o usuario esta enviando para o GOLPHERS' ;

ALTER TABLE GO\_CONTATO ADD CONSTRAINT GO\_CONTATO\_PK PRIMARY KEY ( ID ) ;

**Tabela de Notícias**

CREATE TABLE GO\_NOTICIAS

(

ID NUMBER (16,2) NOT NULL ,

TITULO VARCHAR2 (200) NOT NULL ,

MATERIA VARCHAR2 (4000) NOT NULL ,

IMAGEM BLOB ,

DATA\_MATERIA DATE DEFAULT sysdate NOT NULL ,

STA\_ATIVO VARCHAR2 (1) DEFAULT 'N' NOT NULL ,

GO\_CATEGORIA\_ID NUMBER (16,2) NOT NULL

) ;

ALTER TABLE GO\_NOTICIAS ADD CONSTRAINT GONOTI\_STAT\_CK CHECK ( STA\_ATIVO IN ('N', 'S')) ;

COMMENT ON COLUMN GO\_NOTICIAS.ID IS 'Identifacador tabela TR\_NOTICIAS' ;

COMMENT ON COLUMN GO\_NOTICIAS.TITULO IS 'Titulo da Materia' ;

COMMENT ON COLUMN GO\_NOTICIAS.MATERIA IS 'Noticia';

COMMENT ON COLUMN GO\_NOTICIAS.IMAGEM IS 'Imagem para ilustrar a materia' ;

COMMENT ON COLUMN GO\_NOTICIAS.DATA\_MATERIA IS 'Data que a materia foi publicado.' ;

COMMENT ON COLUMN GO\_NOTICIAS.STA\_ATIVO IS 'Informa se a materia sera exibida no portal ou nao' ;

ALTER TABLE GO\_NOTICIAS ADD CONSTRAINT GO\_NOTICIAS\_PK PRIMARY KEY ( ID ) ;

ALTER TABLE GO\_NOTICIAS ADD CONSTRAINT NOTI\_STA\_ATIVO\_UN UNIQUE ( TITULO , STA\_ATIVO ) ;

**Tabela de Pessoa**

CREATE TABLE GO\_PESSOA

(

ID NUMBER (16,2) NOT NULL ,

NOME\_COMPLETO VARCHAR2 (200) NOT NULL ,

ENDERECO VARCHAR2 (200) NOT NULL ,

EMAIL VARCHAR2 (200 CHAR) ,

DATA\_NASCIMENTO DATE NOT NULL ,

USUARIO VARCHAR2 (100 CHAR) ,

SENHA RAW (200) ,

STA\_ADM VARCHAR2 (1) DEFAULT 'N' NOT NULL

) ;

ALTER TABLE GO\_PESSOA ADD CONSTRAINT GOPESS\_STAD\_CK CHECK ( STA\_ADM IN ('N', 'S')) ;

COMMENT ON TABLE GO\_PESSOA IS 'Tabela para o cadastro de pessoas do sistema, tanto administrados, quanto clientes.' ;

COMMENT ON COLUMN GO\_PESSOA.ID IS 'Identifacador tabela de clientes' ;

COMMENT ON COLUMN GO\_PESSOA.NOME\_COMPLETO IS 'Nome completo da pessoa' ;

COMMENT ON COLUMN GO\_PESSOA.ENDERECO IS 'Endereço completo do cliente.' ;

COMMENT ON COLUMN GO\_PESSOA.EMAIL IS 'Email Cliente' ;

COMMENT ON COLUMN GO\_PESSOA.DATA\_NASCIMENTO IS 'Data de Nascimento ' ;

COMMENT ON COLUMN GO\_PESSOA.USUARIO IS 'Sera usado apenas para a Nac do Gustavo.' ;

COMMENT ON COLUMN GO\_PESSOA.SENHA IS 'Sera usado apenas para a Nac do Gustavo.' ;

COMMENT ON COLUMN GO\_PESSOA.STA\_ADM IS 'Informa se o Usuario e administrador da aplicaçao ou nao, caso esta flag seja igual a S, entao tera acesso a area restrita' ;

ALTER TABLE GO\_PESSOA ADD CONSTRAINT GO\_PESSOA\_PK PRIMARY KEY ( ID ) ;

ALTER TABLE GO\_PESSOA ADD CONSTRAINT GO\_PESSOA\_USUARIO\_UN UNIQUE ( USUARIO ) ;

ALTER TABLE GO\_CONTATO ADD CONSTRAINT GO\_CONTATO\_GO\_PESSOA\_FK FOREIGN KEY ( GO\_PESSOA\_ID ) REFERENCES GO\_PESSOA ( ID ) ;

ALTER TABLE GO\_NOTICIAS ADD CONSTRAINT GO\_NOTICIAS\_GO\_CATEGORIA\_FK FOREIGN KEY ( GO\_CATEGORIA\_ID ) REFERENCES GO\_CATEGORIA ( ID ) ;

**Criação das Sequences**

create sequence seq\_auditoria;

create sequence seq\_noticias;

create sequence seq\_categoria;

create sequence seq\_pessoa;

**Criação das Views**

create or replace view vw\_noticia as

select noti.id

, cate.categoria

, titulo

, materia

, data\_materia

from go\_categoria cate

, go\_noticias noti

where noti.GO\_CATEGORIA\_ID = cate.id

and noti.sta\_ativo = 'S'

and cate.STA\_ATIVO = 'S';

create or replace view vw\_contato as

select pess.NOME\_COMPLETO

, pess.email

, cont.mensagem

from go\_pessoa pess

, go\_contato cont

where cont.GO\_PESSOA\_ID = pess.id

order by cont.DT\_MENSAGEM;

**Scripts PL/SQL**

**Procedure de Auditoria**

Procedure para inserir as auditorias no banco, será chamado apenas dentro das rotinas de update e delete para manter o log

create or replace procedure prc\_insere\_auditoria(p\_tabela go\_auditoria.tabela%type

,p\_alteracao go\_auditoria.tipo\_alteracao%type

,p\_historico go\_auditoria.historico%type

,p\_usuario go\_auditoria.usuario%type) is

begin

insert into go\_auditoria (id

,tabela

,tipo\_alteracao

,historico

,data

,usuario) values (seq\_auditoria.nextval

,p\_tabela

,p\_alteracao

,p\_historico

,sysdate

,p\_usuario);

commit;

exception when others then

Raise\_Application\_Error (SQLCODE, 'Erro ao inserir auditoria!');

rollback;

end prc\_insere\_auditoria;

/

**Package de Categoria**

create or replace package pkg\_categoria is

procedure prc\_insere (p\_categoria go\_categoria.categoria%type

,p\_sta\_ativo go\_categoria.sta\_ativo%type);

procedure prc\_altera (p\_id go\_categoria.id%type

,p\_categoria go\_categoria.categoria%type

,p\_sta\_ativo go\_categoria.sta\_ativo%type

,p\_usuario go\_pessoa.usuario%type);

procedure prc\_deleta (p\_id go\_categoria.id%type

,p\_usuario go\_pessoa.usuario%type);

end pkg\_categoria;

/

create or replace package body pkg\_categoria is

procedure prc\_insere (p\_categoria go\_categoria.categoria%type

,p\_sta\_ativo go\_categoria.sta\_ativo%type) is

begin

if p\_categoria is not null and p\_sta\_ativo is not null then

insert into go\_categoria (id

,categoria

,sta\_ativo) values (seq\_categoria.nextval

,p\_categoria

,p\_sta\_ativo);

commit;

else

Raise\_Application\_Error (-20343, 'Favor preencher todas as informaçoes obrigatorias');

end if;

exception when others then

Raise\_Application\_Error (SQLCODE, 'Ocorreu um erro ao inserir, tente novamente mais tarde');

rollback;

end prc\_insere;

procedure prc\_altera (p\_id go\_categoria.id%type

,p\_categoria go\_categoria.categoria%type

,p\_sta\_ativo go\_categoria.sta\_ativo%type

,p\_usuario go\_pessoa.usuario%type) is

v\_linha\_antiga go\_categoria%rowtype;

v\_historico varchar2(1000);

begin

select \*

into v\_linha\_antiga

from go\_categoria

where id = p\_id;

if p\_categoria != v\_linha\_antiga.categoria then

v\_historico := 'Categoria anterior: '||v\_linha\_antiga.categoria||chr(13);

end if;

if p\_sta\_ativo != v\_linha\_antiga.sta\_ativo then

if p\_sta\_ativo = 'N' then

v\_historico := v\_historico||'Categoria desativada'||chr(13);

elsif p\_sta\_ativo = 'S' then

v\_historico := v\_historico||'Categoria Ativada'||chr(13);

end if;

end if;

update go\_categoria

set categoria = p\_categoria

, sta\_ativo = p\_sta\_ativo

where id = p\_id;

commit;

prc\_insere\_auditoria('GO\_CATEGORIA','UPDATE',v\_historico,p\_usuario);

exception when no\_data\_found then

Raise\_Application\_Error (-20343, 'Categoria não encontrada');

when others then

Raise\_Application\_Error (SQLCODE, 'Ocorreu um erro ao inserir, tente novamente mais tarde');

rollback;

end prc\_altera;

procedure prc\_deleta (p\_id go\_categoria.id%type

,p\_usuario go\_pessoa.usuario%type) is

v\_count\_noticia number;

begin

select count(1)

into v\_count\_noticia

from go\_noticias

where go\_categoria\_id = p\_id;

if v\_count\_noticia != 0 then

Raise\_Application\_Error (-20343, 'Já existem noticias cadastradas com esta categoria, favor deletar as noticias.');

else

delete go\_categoria

where id = p\_id;

commit;

prc\_insere\_auditoria('GO\_CATEGORIA','DELETE',null,p\_usuario);

end if;

exception when others then

Raise\_Application\_Error (SQLCODE, 'Ocorreu um erro ao inserir, tente novamente mais tarde');

rollback;

end prc\_deleta;

end pkg\_categoria;

/

**Bloco Anônimo de Package Categoria**

**Inserir:**

ACCEPT p\_categoria PROMPT 'ENTRE COM A CATEGORIA'

ACCEPT p\_sta\_ativo PROMPT 'ENTRE COM O STATUS'

DECLARE

v\_categoria go\_categoria.categoria%type := '&p\_categoria'

v\_sta\_ativo go\_categoria.sta\_ativo%type := '&p\_sta\_ativo'

BEGIN

pkg\_categoria.prc\_insere(v\_categoria,v\_sta\_ativo);

END;

/

**Atualizar:**

ACCEPT p\_id PROMPT 'ENTRE COM O ID'

ACCEPT p\_categoria PROMPT 'ENTRE COM A CATEGORIA'

ACCEPT p\_sta\_ativo PROMPT 'ENTRE COM O STATUS'

ACCEPT p\_usuario PROMPT 'ENTRE COM O USUARIO'

DECLARE

v\_id go\_categoria.id%type := '&p\_id'

v\_categoria go\_categoria.categoria%type := '&p\_categoria'

v\_sta\_ativo go\_categoria.sta\_ativo%type := '&p\_sta\_ativo'

v\_usuario go\_pessoa.usuario%type := '&p\_usuario'

BEGIN

pkg\_categoria.prc\_altera(v\_id,v\_categoria,v\_sta\_ativo,v\_usuario);

END;

/

**Deletar:**

ACCEPT p\_id PROMPT 'ENTRE COM O ID'

ACCEPT p\_usuario PROMPT 'ENTRE COM O USUARIO'

DECLARE

v\_id go\_categoria.id%type := '&p\_id'

v\_usuario go\_pessoa.usuario%type := '&p\_usuario'

BEGIN

pkg\_categoria.prc\_deleta(v\_id,v\_usuario);

END;

/

**Package Login**

Package com função de login para utilização em API e critografia para segurança de senha

create or replace package pkg\_login is

function criptografia (p\_valor varchar) return raw;

function login (p\_usuario go\_pessoa.usuario%type,p\_senha go\_pessoa.senha%type) return boolean;

end;

/

create or replace package body pkg\_login is

function criptografia (p\_valor varchar) return raw is

senha\_cripto raw(32) := null;

begin

senha\_cripto := rawtohex(dbms\_obfuscation\_toolkit.md5(input => utl\_raw.cast\_to\_raw(p\_valor)));

return nvl(senha\_cripto,'');

end criptografia;

function login (p\_usuario go\_pessoa.usuario%type,p\_senha go\_pessoa.senha%type) return boolean is

v\_senha\_cripto raw := criptografia(p\_senha);

v\_count\_user number;

begin

select count(1)

into v\_count\_user

from go\_pessoa

where usuario = p\_usuario

and senha = v\_senha\_cripto;

if v\_count\_user = 0 then

return false;

elsif v\_count\_user = 1 then

return true;

else

Raise\_Application\_Error (-20343, 'Foi localizado mais de um cadastro com o mesmo usuario e senha.');

end if;

end login;

end pkg\_login;

/

**Package Pessoa**

create or replace package pkg\_pessoa is

procedure prc\_insere(p\_nome go\_pessoa.nome\_completo%type,p\_end go\_pessoa.endereco%type,p\_email go\_pessoa.email%type, p\_data\_nascimento go\_pessoa.data\_nascimento%type, p\_usuario go\_pessoa.usuario%type, p\_senha go\_pessoa.senha%type);

procedure envia\_pergunta(p\_mensagem go\_contato.mensagem%type, p\_id\_usario go\_pessoa.id%type);

end pkg\_pessoa;

/

create or replace package body pkg\_pessoa is

procedure prc\_insere(p\_nome go\_pessoa.nome\_completo%type,p\_end go\_pessoa.endereco%type,p\_email go\_pessoa.email%type, p\_data\_nascimento go\_pessoa.data\_nascimento%type, p\_usuario go\_pessoa.usuario%type, p\_senha go\_pessoa.senha%type) is

v\_senha\_criptografa raw := pkg\_login.criptografia(p\_senha);

begin

if p\_nome is not null and p\_end is not null and p\_data\_nascimento is not null and p\_usuario is not null and v\_senha\_criptografa is not null then

insert into go\_pessoa(id

,nome\_completo

,endereco

,email

,data\_nascimento

,usuario

,senha) values (seq\_pessoa.nextval

,p\_nome

,p\_end

,p\_email

,p\_data\_nascimento

,p\_usuario

,v\_senha\_criptografa);

commit;

else

Raise\_Application\_Error (-20343, 'Favor preencher todas as informaçoes obrigatorias');

end if;

exception when others then

rollback;

Raise\_Application\_Error (SQLERROR, 'Ocorreu um erro');

end prc\_insere;

procedure envia\_pergunta (p\_mensagem go\_contato.mensagem%type, p\_id\_usario go\_pessoa.id%type) is

begin

if p\_mensagem is not null then

insert into go\_contato(id

,dt\_mensagem

,mensagem

,go\_pessoa\_id) values (seq\_pessoa.nextval

,sysdate

,p\_mensagem

,p\_id\_usario);

commit;

else

Raise\_Application\_Error (-20343, 'Favor preencher enviar a mensagem');

end if;

exception when others then

rollback;

Raise\_Application\_Error (SQLERROR, 'Ocorreu um erro');

end;

end pkg\_pessoa;

/

**Bloco Anônimo para Pessoa**

**Insere:**

ACCEPT p\_nome PROMPT 'ENTRE COM O NOME'

ACCEPT p\_endereco PROMPT 'ENTRE COM O ENDERECO'

ACCEPT p\_email PROMPT 'ENTRE COM O EMAIL'

ACCEPT p\_data\_nascimento PROMPT 'ENTRE COM O DATA DE NASCIMENTO'

ACCEPT p\_usuario PROMPT 'ENTRE COM O USUARIO'

ACCEPT p\_senha PROMPT 'ENTRE COM A SENHA'

DECLARE

V\_nome go\_pessoa.nome\_completo%type :='&p\_nome';

V\_end go\_pessoa.endereco%type :='&p\_endereco';

V\_email go\_pessoa.email%type :='&p\_email';

V\_data\_nascimento go\_pessoa.data\_nascimento%type :='&p\_data\_nascimento';

V\_usuario go\_pessoa.usuario%type :='&p\_usuario';

V\_senha go\_pessoa.senha%type :='&p\_senha';

BEGIN

pkg\_pessoa.prc\_insere(v\_nome,v\_end,v\_email,v\_data\_nascimento,v\_usuario,v\_senha);

END;

/

**Enviar Pergunta:**

ACCEPT p\_mensagem PROMPT 'ENTRE COM A MENSAGEM'

ACCEPT p\_id PROMPT 'ENTRE COM O ID DO USUARIO'

DECLARE

V\_MENSAGEM go\_contato.mensagem%type :='&p\_mensagem';

v\_id go\_pessoa.id%type :='&p\_id';

BEGIN

pkg\_pessoa.envia\_pergunta(v\_mensagem,v\_id);

END;

/

**Package de Noticias**

create or replace package pkg\_noticias is

procedure prc\_insere (p\_titulo go\_noticias.titulo%type,p\_materia go\_noticias.materia%type,p\_imagem go\_noticias.imagem%type,p\_sta\_ativo go\_noticias.sta\_ativo%type,p\_id\_categoria go\_noticias.go\_categoria\_id%type);

procedure prc\_atualiza (p\_id go\_noticias.id%type,p\_titulo go\_noticias.titulo%type,p\_materia go\_noticias.materia%type,p\_imagem go\_noticias.imagem%type,p\_sta\_ativo go\_noticias.sta\_ativo%type,p\_id\_categoria go\_categoria.id%type,p\_usuario go\_pessoa.usuario%type);

procedure prc\_deleta (p\_id\_noticias go\_noticias.id%type, p\_usuario go\_pessoa.usuario%type);

end pkg\_noticias;

/

create or replace package body pkg\_noticias is

procedure prc\_insere (p\_titulo go\_noticias.titulo%type,p\_materia go\_noticias.materia%type,p\_imagem go\_noticias.imagem%type,p\_sta\_ativo go\_noticias.sta\_ativo%type,p\_id\_categoria go\_noticias.go\_categoria\_id%type) is

begin

if p\_titulo is not null and p\_materia is not null and p\_sta\_ativo is not null then

insert into go\_noticias (id

,titulo

,materia

,imagem

,data\_materia

,sta\_ativo

,go\_categoria\_id) values(seq\_noticias.nextval

,p\_titulo

,p\_materia

,p\_imagem

,sysdate

,p\_sta\_ativo

,p\_id\_categoria);

commit;

else

Raise\_Application\_Error (-20343, 'Favor preencher todas as informaçoes obrigatorias');

end if;

exception when others then

rollback;

Raise\_Application\_Error (SQLERROR, 'Ocorreu um erro');

end prc\_insere;

**Bloco Anônimo de Noticias**

procedure prc\_atualiza (p\_id go\_noticias.id%type,p\_titulo go\_noticias.titulo%type,p\_materia go\_noticias.materia%type,p\_imagem go\_noticias.imagem%type,p\_sta\_ativo go\_noticias.sta\_ativo%type,p\_id\_categoria go\_categoria.id%type,p\_usuario go\_pessoa.usuario%type) IS

v\_linha\_antiga go\_noticias%rowtype;

v\_historico varchar2(4000);

v\_categoria\_antiga go\_categoria.categoria%type;

begin

if p\_titulo is not null and p\_materia is not null and p\_sta\_ativo is not null then

select \*

into v\_linha\_antiga

from go\_noticias

where id = p\_id;

if v\_linha\_antiga.titulo != p\_titulo then

v\_historico := 'Titulo Antigo: '||p\_titulo||chr(13);

end if;

if v\_linha\_antiga.materia != p\_materia then

v\_historico := v\_historico||'Materia Antiga: '||p\_materia||chr(13);

end if;

if v\_linha\_antiga.sta\_ativo != p\_sta\_ativo then

if p\_sta\_ativo = 'S' then

v\_historico := v\_historico||'Materia ativada '||chr(13);

end if;

if p\_sta\_ativo = 'N' then

v\_historico := v\_historico||'Materia desativada '||chr(13);

end if;

end if;

if v\_linha\_antiga.go\_categoria\_id != p\_id\_categoria then

begin

select categoria

into v\_categoria\_antiga

from go\_categoria

where id = v\_linha\_antiga.go\_categoria\_id;

exception when no\_data\_found then

Raise\_Application\_Error (-20343, 'Categoria antiga nao encontrada, favor verificar o cadastro');

when too\_many\_rows then

Raise\_Application\_Error (-20343, 'Categoria antiga duplicada, favor verificar o cadastro');

end;

v\_historico := v\_historico||'Categoria antiga: '||v\_categoria\_antiga;

end if;

update go\_noticias

set titulo = p\_titulo

, materia = p\_materia

, imagem = p\_imagem

, sta\_ativo = p\_sta\_ativo

, go\_categoria\_id = p\_id\_categoria

where id = p\_id;

commit;

prc\_insere\_auditoria('GO\_NOTICIAS','UPDATE',v\_historico,p\_usuario);

else

Raise\_Application\_Error (-20343, 'Favor preencher todas as informaçoes obrigatorias');

end if;

exception when others then

rollback;

Raise\_Application\_Error (SQLERROR, 'Ocorreu um erro');

end prc\_atualiza;

procedure prc\_deleta (p\_id\_noticias go\_noticias.id%type, p\_usuario go\_pessoa.usuario%type) is

begin

delete go\_noticias

where id = p\_id\_noticias;

prc\_insere\_auditoria('GO\_NOTICIAS','DELETE',null,p\_usuario);

commit;

exception when others then

rollback;

Raise\_Application\_Error (SQLERROR, 'Ocorreu um erro');

end prc\_deleta;

end pkg\_noticias;

/