

Pacote de funções - PostgreSQL

Criando o SCHEMA PKG_EMA

```
CREATE SCHEMA pkg_ema AUTHORIZATION postgres;
```

RETORNA COLUNA GRADE

```
-- DROP FUNCTION pkg_ema.retorna_coluna_grade(int4, int4, int4, int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_coluna_grade(xidprocesso integer, xidatividade
integer, xidformulario integer, xidgrade integer, xidvalor integer)
RETURNS character varying
LANGUAGE plpgsql
AS $function$
declare
    leitura varchar(4000);
begin
    select
        resposta into leitura
    from public.crm_processo_grade_valor
    where idprocesso = xidprocesso
        and idformulario = xidformulario
        and idatividade = xidatividade
        and idgrade = xidgrade
        and idvalor = xidvalor
        and idrepeticao = (select max(idrepeticao)
            from crm_processo_grade_valor xx
            where xx.idatividade = crm_processo_grade_valor.idatividade
                and xx.idprocesso = crm_processo_grade_valor.idprocesso
                and xx.idformulario = crm_processo_grade_valor.idformulario);
    return(trim(leitura));
end;
$function$
;
```

RETORNA COLUNA GRADE GLOB

```
-- DROP FUNCTION pkg_ema.retorna_coluna_grade_clob(int4, int4, int4, int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_coluna_grade_clob(xidprocesso integer, xidatividade
integer, xidformulario integer, xidgrade integer, xidvalor integer)
RETURNS text
LANGUAGE plpgsql
AS $function$
declare
    leitura text;
begin
    select
        resposta into leitura
    from public.crm_processo_grade_valor
    where idprocesso = xidprocesso
        and idformulario = xidformulario
        and idatividade = xidatividade
        and idgrade = xidgrade
        and idvalor = xidvalor
        and idrepeticao = (select max(idrepeticao)
            from crm_processo_grade_valor xx
            where xx.idatividade = crm_processo_grade_valor.idatividade
                and xx.idprocesso = crm_processo_grade_valor.idprocesso
                and xx.idformulario = crm_processo_grade_valor.idformulario);
    return(trim(leitura));
end;
$function$
;
```

RETORNA COLUNA GRADE DATA

```
-- DROP FUNCTION pkg_ema.retorna_coluna_grade_data(int4, int4, int4, int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_coluna_grade_data(xidprocesso integer, xidatividade
integer, xidformulario integer, xidgrade integer, xidvalor integer)
RETURNS date
LANGUAGE plpgsql
AS $function$
declare
    leitura date;
begin
```

```

select respostadatahora into leitura
from public.crm_processo_grade_valor
where idprocesso = xidprocesso
and idformulario= xidformulario
and idatividade = xidatividade
and idgrade = xidgrade
and idvalor = xidvalor
and idrepeticao = (select max(idrepeticao)
from crm_processo_grade_valor xx
where xx.idatividade = crm_processo_grade_valor.idatividade
and xx.idprocesso = crm_processo_grade_valor.idprocesso
and xx.idformulario = crm_processo_grade_valor.idformulario);
return(leitura); --fazer o trim depois

end;
$function$
;

```

RETORNA COLUNA GRADE DATAHORA

```

-- DROP FUNCTION pkg_ema.retorna_coluna_grade_datahr(int4, int4, int4, int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_coluna_grade_datahr(xidprocesso integer,
xidatividade integer, xidformulario integer, xidgrade integer, xidvalor integer)
RETURNS timestamp without time zone
LANGUAGE plpgsql
AS $function$
declare
leitura timestamp;
begin
select respostadatahora into leitura
from public.crm_processo_grade_valor
where idprocesso = xidprocesso
and idformulario= xidformulario
and idatividade = xidatividade
and idgrade = xidgrade
and idvalor = xidvalor
and idrepeticao = (select max(idrepeticao)
from crm_processo_grade_valor xx
where xx.idatividade = crm_processo_grade_valor.idatividade

```

```

        and xx.idprocesso = crm_processo_grade_valor.idprocesso
        and xx.idformulario = crm_processo_grade_valor.idformulario);
return(leitura);

end;
$function$
;

```

RETORNA COLUNA GRADE INTEIRO

```

-- DROP FUNCTION pkg_ema.retorna_coluna_grade_int(int4, int4, int4, int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_coluna_grade_int(xidprocesso integer, xidatividade
integer, xidformulario integer, xidgrade integer, xidvalor integer)
RETURNS integer
LANGUAGE plpgsql
AS $function$
declare
    leitura int4;

begin
    /*leitura := 0 ;
select (select to_number(decode(respostainteiro, '.',0,respostainteiro)) from dual)*/

select respostainteiro into leitura
from public.crm_processo_grade_valor
where idprocesso = xidprocesso
and idformulario = xidformulario
and idatividade = xidatividade
and idgrade = xidgrade
and idvalor = xidvalor
and idrepeticao = (select max(idrepeticao)
from crm_processo_grade_valor xx
where xx.idatividade = crm_processo_grade_valor.idatividade
and xx.idprocesso = crm_processo_grade_valor.idprocesso);

return(leitura);

end;
$function$

```

```
;
```

RETORNA COLUNA GRADE TEXTO

```
-- DROP FUNCTION pkg_ema.retorna_coluna_grade_texto(int4, int4, int4, int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_coluna_grade_texto(xidprocesso integer,
xidatividade integer, xidformulario integer, xidgrade integer, xidvalor integer)
RETURNS character varying
LANGUAGE plpgsql
AS $function$
declare
    leitura varchar(4000);
begin
    select
        resposta into leitura
    from public.crm_processo_grade_valor
    where idprocesso = xidprocesso
        and idformulario = xidformulario
        and idatividade = xidatividade
        and idgrade = xidgrade
        and idvalor = xidvalor
        and idrepeticao = (select max(idrepeticao)
            from crm_processo_grade_valor xx
            where xx.idatividade = crm_processo_grade_valor.idatividade
            and xx.idprocesso = crm_processo_grade_valor.idprocesso
            and xx.idformulario = crm_processo_grade_valor.idformulario);
    return(trim(leitura));
end;
$function$
;
```

RETORNA COLUNA GRADE VALOR

```
-- DROP FUNCTION pkg_ema.retorna_coluna_grade_valor(int4, int4, int4, int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_coluna_grade_valor(xidprocesso integer,
xidatividade integer, xidformulario integer, xidgrade integer, xidvalor integer)
RETURNS numeric
LANGUAGE plpgsql
```

```

AS $function$
declare
  leitura numeric(15,2);
begin
  select respostavalor into leitura
  from public.crm_processo_grade_valor
  where idprocesso = xidprocesso
  and idformulario = xidformulario
  and idatividade = xidatividade
  and idgrade = xidgrade
  and idvalor = xidvalor
  and idrepeticao = (select max(idrepeticao)
                    from crm_processo_grade_valor xx
                    where xx.idatividade = crm_processo_grade_valor.idatividade
                    and xx.idprocesso = crm_processo_grade_valor.idprocesso);
  return(leitura);
end;
$function$
;

```

RETORNA LISTA

```

-- DROP FUNCTION pkg_ema.retorna_lista(text, bpchar);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_lista(in_lista text, delimitador character)
  RETURNS TABLE(out_id integer, out_str text)
  LANGUAGE plpgsql
AS $function$
declare
  dados record;
begin
  IN_LISTA = REPLACE(REPLACE(IN_LISTA, '(', ''), ')', '');

  IF DELIMITADOR = '' OR DELIMITADOR IS NULL THEN DELIMITADOR = ','; END IF;
  IF IN_LISTA <> '.' and IN_LISTA is not null and IN_LISTA <> '' and (IN_LISTA not like '%/'
and IN_LISTA not like '%/') THEN
  for dados in (select unnest(string_to_array(IN_LISTA, DELIMITADOR)) LISTA)
  LOOP
  BEGIN
  OUT_ID = cast(dados.lista as integer);

```

```

        OUT_STR = dados.lista;
RETURN NEXT;

EXCEPTION
WHEN OTHERS then
    OUT_ID = 0;
OUT_STR = dados.lista;
    RETURN NEXT;
END;
END LOOP;
ELSE
OUT_ID = 0;
    OUT_STR = '0';
    RETURN NEXT;
END IF;

end;
$function$
;

```

RETORNA VALOR VARIAVEL

```

-- DROP FUNCTION pkg_ema.retorna_valor_variavel(int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_valor_variavel(xidprocesso integer, xidvariavel
integer)
    RETURNS character varying
    LANGUAGE plpgsql
AS $function$
    declare
    valor varchar(5000);
    begin
    select valoratual into valor
    from public.crm_processo_variavel
    where.crm_processo_variavel.idprocesso = xidprocesso
    and.crm_processo_variavel.idvariavel = xidvariavel ;
    return valor;
    end;
$function$
;

```

RETORNA VALOR VARIAVEL BLOB

```
-- DROP FUNCTION pkg_ema.retorna_valor_variavel_blob(int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_valor_variavel_blob(xidprocesso integer,
xidvariavel integer)
  RETURNS text
  LANGUAGE plpgsql
AS $function$
  declare
  valor text;
  begin
  select valorblob into valor
  from public.crm_processo_variavel
  where.crm_processo_variavel.idprocesso = xidprocesso
  and.crm_processo_variavel.idvariavel = xidvariavel ;
  return trim(valor);
  end;
$function$
;
```

RETORNA VALOR VARIAVEL CLOB

```
-- DROP FUNCTION pkg_ema.retorna_valor_variavel_clob(int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_valor_variavel_clob(xidprocesso integer,
xidvariavel integer)
  RETURNS text
  LANGUAGE plpgsql
AS $function$
  declare
  valor text;
  begin
  select valoratual into valor
  from public.crm_processo_variavel
  where.crm_processo_variavel.idprocesso = xidprocesso
  and.crm_processo_variavel.idvariavel = xidvariavel ;
  return trim(valor);
  end;
$function$
```

```
;
```

RETORNA VALOR VARIAVEL DATA

```
-- DROP FUNCTION pkg_ema.retorna_valor_variavel_data(int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_valor_variavel_data(xidprocesso integer,
xidvariavel integer)
  RETURNS date
  LANGUAGE plpgsql
AS $function$
  declare
    valor date;
  begin
    if (xidprocesso > 0 and xidvariavel > 0) then
      begin
        select case when valoratual <> '.' then to_date(valoratual,'dd/mm/yyyy HH24:MI')
          else to_date('01-01-1900','dd/mm/yyyy HH24:MI') end into valor
        from public.crm_processo_variavel
        where crm_processo_variavel.idprocesso = xidprocesso
        and crm_processo_variavel.idvariavel = xidvariavel ;
      end;
    else
      valor := to_date('01/01/1900','dd/mm/yyyy HH24:MI');
    end if;
    return valor;
  end;
$function$
;
```

RETORNA VALOR VARIAVEL DATAHORA

```
-- DROP FUNCTION pkg_ema.retorna_valor_variavel_datahr(int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_valor_variavel_datahr(xidprocesso integer,
xidvariavel integer)
  RETURNS timestamp without time zone
  LANGUAGE plpgsql
AS $function$
  declare
```

```

    valor timestamp;
begin
if (xidprocesso > 0 and xidvariavel > 0) then
begin
select case when valoratual <> '.' then to_timestamp(valoratual,'dd/mm/yyyy HH24:MI')
    else to_timestamp('01-01-1900','dd/mm/yyyy HH24:MI') end into valor
from public.crm_processo_variavel
where.crm_processo_variavel.idprocesso = xidprocesso
and.crm_processo_variavel.idvariavel = xidvariavel ;
end;
else
valor := to_timestamp('01/01/1900','dd/mm/yyyy HH24:MI');
end if;
return valor;
end;
$function$
;

```

RETORNA VALOR VARIÁVEL INTEIRO

```

-- DROP FUNCTION pkg_ema.retorna_valor_variavel_int(int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_valor_variavel_int(xidprocesso integer, xidvariavel
integer)
RETURNS integer
LANGUAGE plpgsql
AS $function$
declare
valor int4;
begin
select cast(case when valoratual = '.' then '0' else valoratual end as int4)
into valor
from public.crm_processo_variavel
where.crm_processo_variavel.idprocesso = xidprocesso
and.crm_processo_variavel.idvariavel = xidvariavel ;
return valor;
end;
$function$
;

```

RETORNA VALOR VARIAVEL TEXTO

```
-- DROP FUNCTION pkg_ema.retorna_valor_variavel_texto(int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_valor_variavel_texto(xidprocesso integer,
xidvariavel integer)
RETURNS character varying
LANGUAGE plpgsql
AS $function$
declare
valor varchar(5000);
begin
select valoratual into valor
from public.crm_processo_variavel
where crm_processo_variavel.idprocesso = xidprocesso
and crm_processo_variavel.idvariavel = xidvariavel ;
return valor;
end;
$function$
;
```

RETORNA VALOR VARIAVEL VALOR

```
-- DROP FUNCTION pkg_ema.retorna_valor_variavel_valor(int4, int4);

CREATE OR REPLACE FUNCTION pkg_ema.retorna_valor_variavel_valor(xidprocesso integer,
xidvariavel integer)
RETURNS numeric
LANGUAGE plpgsql
AS $function$
declare
valor decimal(15,2) ;
begin
select replace(replace(case when valoratual = '.' then '0' else valoratual end, '.', ''),
',', '.')
into valor
from public.crm_processo_variavel
where crm_processo_variavel.idprocesso = xidprocesso
and crm_processo_variavel.idvariavel = xidvariavel ;
return valor;
end;
```

```
end;  
$function$  
;
```

EXEMPLO RETORNO DE GRADE

```
SELECT PKG_EMA.RETORNA_COLUNA_GRADE_INT(X.IDPROCESSO, X.IDATIVIDADE, X.IDFORMULARIO, 1,  
X.IDVALOR) AS NOMECAMPO1,  
       PKG_EMA.RETORNA_COLUNA_GRADE_INT(X.IDPROCESSO, X.IDATIVIDADE, X.IDFORMULARIO, 2,  
X.IDVALOR) AS NOMECAMPO2,  
       PKG_EMA.RETORNA_COLUNA_GRADE_INT(X.IDPROCESSO, X.IDATIVIDADE, X.IDFORMULARIO, 3,  
X.IDVALOR) AS NOMECAMPO3,  
       PKG_EMA.RETORNA_COLUNA_GRADE_VALOR(X.IDPROCESSO, X.IDATIVIDADE, X.IDFORMULARIO, 4,  
X.IDVALOR) AS NOMECAMPO4,  
       0/*IDPROCESSO*/ AS IDPROCESSO  
FROM CRM_PROCESSO_GRADE_VALOR X  
WHERE IDPROCESSO = 0/*IDPROCESSO*/  
      AND IDATIVIDADE = 3/*IDATIVIDADE*/  
      AND IDFORMULARIO = 34/*IDFORMULARIO*/  
      AND IDREPETICAO = (SELECT MAX(IDREPETICAO)  
                          FROM CRM_PROCESSO_GRADE_VALOR  
                          WHERE IDPROCESSO = X.IDPROCESSO  
                          AND IDATIVIDADE = X.IDATIVIDADE)  
  
AND IDGRADE = 1
```

Revision #6

Created 7 July 2026 20:27:06 by Nicolly Andrielly

Updated 7 July 2026 20:49:21 by Nicolly Andrielly