

Código postgresQL de parte de la función _importar_archivo_RRHH:

```
CREATE OR REPLACE FUNCTION f_importar_archivo_RRHH(ruta character varying, nombre
character varying) -- Falta agregar parámetros de período
  RETURNS character varying AS
$BODY$
DECLARE
  v_ruta character varying;
  v_nombre character varying;
  v_rutaMasNombre character varying;
  v_filas Integer;
  v_tempPersonal RECORD;
BEGIN
  v_ruta := ruta;
  v_nombre := nombre;
  v_rutaMasNombre = $1||'\'||$2;
  DROP TABLE IF EXISTS tmp_rrhh;
  CREATE TEMP TABLE tmp_hh( column1 varchar(200)); -- ver que columnas agregar a la
tabla temp para facilitar la importacion
  execute 'copy tmp_personal from ''' ||v_rutaMasNombre||''' CSV HEADER DELIMITER
E''\t''';

--Agregar la lógica necesaria
  return 'Importación de archivo de personal: IMPORTACION COMPLETA. Archivo: ' ||
v_nombre || ' - ' ||
to_char(v_filas,'999999') || ' filas - ' || to_char(current_timestamp,'DD/MM/YYYY -
HH24:MI:SS');
END;
$BODY$
LANGUAGE plpgsql VOLATILE
```

Invocación de la función:

```
select f_importar_archivo_RRHH('c:\tmp','Archivo_descuentos_RRHH.csv')
```

Funciones auxiliares

Para validar los tipos de datos se definieron estas funciones. Si bien puede validarse de otro modo, la utilización de las siguientes funciones puede clarificar la codificación.

-- Funciones auxiliares

```
create or replace function f_esEntero (text) returns boolean as $$
begin
  if ($1 is null) then
    return false;
  end if;
  perform $1::Integer;
  return true;
exception when others then
  return false;
end;
$$ language plpgsql;

create or replace function f_esFecha (text) returns boolean as $$
begin
  if ($1 is null) then
    return false;
  end if;
  perform $1::date;
  return true;
exception when others then
  return false;
end;
```

-

```
CREATE OR REPLACE FUNCTION is_number(prm_str text)
  RETURNS boolean AS
$BODY$
DECLARE
  v_return BOOLEAN;
BEGIN
  IF regexp_matches(prm_str,E'^-*[[:digit:]]+\.[[:digit:]]+$') is not null
  THEN
    v_return = TRUE;
  ELSE
    v_return = FALSE;
  END IF;
  RETURN v_return;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION is_number(text)
  OWNER TO postgres;
$$ language plpgsql;
```