--Equipo 3 IDFT301

/\*Para saber el número de tareas e información sobre las mismas según el lugar, esto para que el usuario pueda dar prioridad a la visita de un lugar con respecto a otro\*/

--fav y with

select t.\*, count(\*) over(partition by id\_place\_id) as cant from api\_task t order by cant desc;

with cant\_by\_place as (select id\_place\_id, count(\*) as cant

from api\_task group by id\_place\_id)

select t.\*, cbp.cant from cant\_by\_place cbp, api\_task t

where t.id\_place\_id = cbp.id\_place\_id order by cant desc;

/\*Antes de insertar una tarea, verifica si la fecha de finalización es anterior a la fecha actual, si lo es, la tarea no se inserta porque su fecha ya expiró\*/

--trigger

create function deletePassDate() returns trigger as

$$

begin

if new.end\_datetime < current\_date then

return null;

else

return new;

end if;

end

$$

language 'plpgsql';

create trigger select\_tasks before insert on api\_task for each row

execute procedure deletePassDate();

/\*Elimina las tareas que ya han caducado según su fecha de finalización y devuelve las que siguen vigentes\*/

--cursor

create or replace function select\_task() returns setof api\_task as

$$

declare

var api\_task%rowtype;

cur cursor for (select \* from api\_task);

begin

open cur;

loop

fetch cur into var;

exit when not found;

if var.end\_datetime<current\_date then

delete from api\_task where current of cur;

else

return next var;

end if;

end loop;

close cur;

end

$$

language 'plpgsql'