

Universidad de Costa Rica Sede de Guanacaste

Manual creación y uso de triggers para insertar

Administración de Bases de Datos IF5100

25 de junio de 2021

Contenidos

1	Preparando el ambiente de trabajo	7
1.1	Iniciando la máquina virtual	8
1.2	Conexión mediante PuTTY	9
2	Ingreso a PgAdmin	10
2.1	Ingreso a PgAdmin	11
2.2	Tablas en Base de datos	12
2.3	Triggers en la base de datos	13
3	Triggers Insertar	14
3.1	Tabla pais	15
3.1.1	Trigger que llama a la función que registra en la Tabla Bitácora País	15
3.1.2	Inserción y materialización del Trigger en la bitácora país	16
3.1.3	<i>SELECT BitacoraPais</i>	17
3.2	Tabla provincia	18
3.2.1	Trigger que llama a la función que registra en la Tabla Bitácora Provincia	18
3.2.2	Inserción y materialización del Trigger en la bitácora provincia	19
3.2.3	<i>SELECT BitacoraProvincia</i>	20
3.3	Tabla canton	21
3.3.1	Trigger que llama a la función que registra en la Tabla Bitácora Canton	21
3.3.2	Inserción y materialización del Trigger en la bitácora canton	22
3.3.3	<i>SELECT BitacoraCanton</i>	23
3.4	Tabla distrito	24
3.4.1	Trigger que llama a la función que registra en la Tabla Bitácora Distrito	24
3.4.2	Inserción y materialización del Trigger en la bitácora distrito	25
3.4.3	<i>SELECT BitacoraDistrito</i>	26
3.5	Tabla barrio	27
3.5.1	Trigger que llama a la función que registra en la Tabla Bitácora Barrio	27
3.5.2	Inserción y materialización del Trigger en la bitácora barrio	29
3.5.3	<i>SELECT BitacoraBarrio</i>	30

3.6	Tabla empleados	31
3.6.1	Trigger que llama a la función que registra en la Tabla Bitácora Empleados	31
3.6.2	Inserción y materialización del Trigger en la bitácora empleados	33
3.6.3	<i>SELECT BitacoraEmpleados</i>	35
3.7	Tabla contacto	37
3.7.1	Trigger que llama a la función que registra en la Tabla Bitácora Contacto	37
3.7.2	Inserción y materialización del Trigger en la bitácora contactos	38
3.7.3	<i>SELECT BitacoraContacto</i>	40
3.8	Tabla guia	41
3.8.1	Trigger que llama a la función que registra en la Tabla Bitácora Guia	41
3.8.2	Inserción y materialización del Trigger en la bitácora guia	43
3.8.3	<i>SELECT BitacoraGuia</i>	44
3.9	Tabla duracion_recorrido	45
3.9.1	Trigger que llama a la función que registra en la Tabla Bitácora Duracion_recorrido	45
3.9.2	Inserción y materialización del Trigger en la bitácora duracion recorrido	47
3.9.3	<i>SELECT BitacoraDuraccionRecorrido</i>	48
3.10	Tabla itinerarios	49
3.10.1	Trigger que llama a la función que registra en la Tabla Bitácora Itinerarios	49
3.10.2	Inserción y materialización del Trigger en la bitácora itinerario	51
3.10.3	<i>SELECT BitacoraItinerarios</i>	52
3.11	Tabla asignaciones	53
3.11.1	Trigger que llama a la función que registra en la Tabla Bitácora Asignaciones	53
3.11.2	Inserción y materialización del Trigger en la bitácora asignaciones	54
3.11.3	<i>SELECT BitacoraAsignaciones</i>	55
3.12	Tabla continente	56
3.12.1	Trigger que llama a la función que registra en la Tabla Bitácora Continente	56
3.12.2	Inserción y materialización del Trigger en la bitácora continente	57
3.12.3	<i>SELECT BitacoraContinente</i>	58
3.13	Tabla cuidadores	59

3.13.1	Trigger que llama a la función que registra en la Tabla Bitácora Cuidadores	59
3.13.2	Inserción y materialización del Trigger en la bitácora cuidador	60
3.13.3	<i>SELECT BitacoraCuidadores</i>	61
3.14	Tabla cuidador_especie	62
3.14.1	Trigger que llama a la función que registra en la Tabla Bitácora Cuidador Especie	62
3.14.2	Inserción y materialización del Trigger en la bitácora cuidador especie	63
3.14.3	<i>SELECT BitacoraCuidadorEspecie</i>	64
3.15	Tabla descripcion_especie	65
3.15.1	Trigger que llama a la función que registra en la Tabla Bitácora Descripcion Especie	65
3.15.2	Inserción y materialización del Trigger en la bitácora descripcion especie	66
3.15.3	<i>SELECT BitacoraDescripcion_especie</i>	67
3.16	Tabla especie	68
3.16.1	Trigger que llama a la función que registra en la Tabla Bitácora Especie	68
3.16.2	Inserción y materialización del Trigger en la bitácora especie	69
3.16.3	<i>SELECT BitacoraEspecie</i>	70
3.17	Tabla especie_habitat	71
3.17.1	Trigger que llama a la función que registra en la Tabla Bitácora Especie Habitat	71
3.17.2	Inserción y materialización del Trigger en la bitácora especie habitat	72
3.17.3	<i>SELECT BitacoraEspecie_Habitat</i>	73
3.18	Tabla habitat	74
3.18.1	Trigger que llama a la función que registra en la Tabla Bitácora Habitat	74
3.18.2	Inserción y materialización del Trigger en la bitácora habitat	75
3.18.3	<i>SELECT BitacoraHabitat</i>	76
3.19	Tabla lista_dezonas_visitadas	77
3.19.1	Trigger que llama a la función que registra en la Tabla Bitácora Lista de zonas visitadas	77
3.19.2	Inserción y materialización del Trigger en la bitácora lista de zonas visitadas	78
3.19.3	<i>SELECT BitacoraLista_dezonas_visitadas</i>	79
3.20	Tabla tipo_habitat	80
3.20.1	Trigger que llama a la función que registra en la Tabla Bitácora Tipo Habitat	80

3.20.2	Inserción y materialización del Trigger en la bitácora tipo de habitat	81
3.20.3	<i>SELECT BitacoraTipo_habitat</i>	82
3.21	Tabla zona	83
3.21.1	Trigger que llama a la función que registra en la Tabla Bitácora Zona	83
3.21.2	Inserción y materialización del Trigger en la bitácora tipo de zona	84
3.21.3	<i>SELECT BitacoraZona</i>	85

Página en blanco

1 Preparando el ambiente de trabajo

1.1 Iniciando la máquina virtual

Procedemos a iniciar una máquina virtual para la realización de prueba de vistas materializadas

```
Ubuntu 20.04.2 LTS bdatos tty6
bdatos login: gnu
Password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-74-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sat 12 Jun 2021 12:34:13 AM CST

System load:  0.12           Users logged in:      1
Usage of /:   17.9% of 38.6GB IPv4 address for docker0: 172.17.0.1
Memory usage: 47%           IPv4 address for enp0s3:  10.0.2.15
Swap usage:   0%            IPv4 address for enp0s8:  192.168.56.101
Processes:    145

 * Super-optimized for small spaces - read how we shrank the memory
   footprint of MicroK8s to make it the smallest full K8s around.

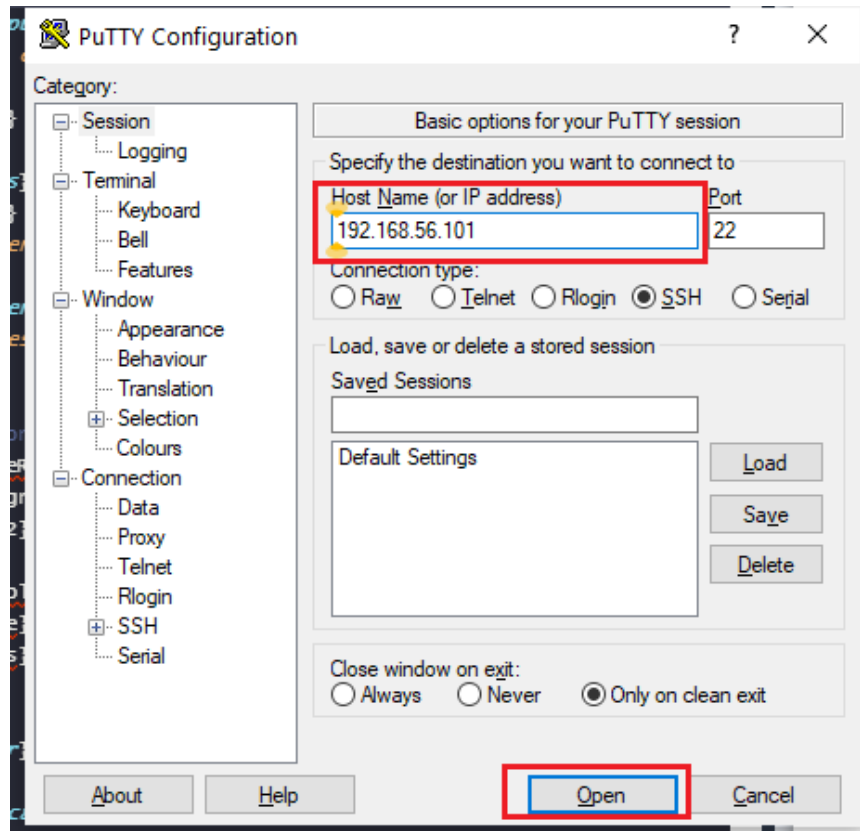
   https://ubuntu.com/blog/microk8s-memory-optimisation

91 updates can be installed immediately.
23 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Last login: Sat Jun 12 00:18:12 CST 2021 from 192.168.56.1 on pts/0
gnu@bdatos:~$ _
```


1.2 Conexión mediante PuTTY

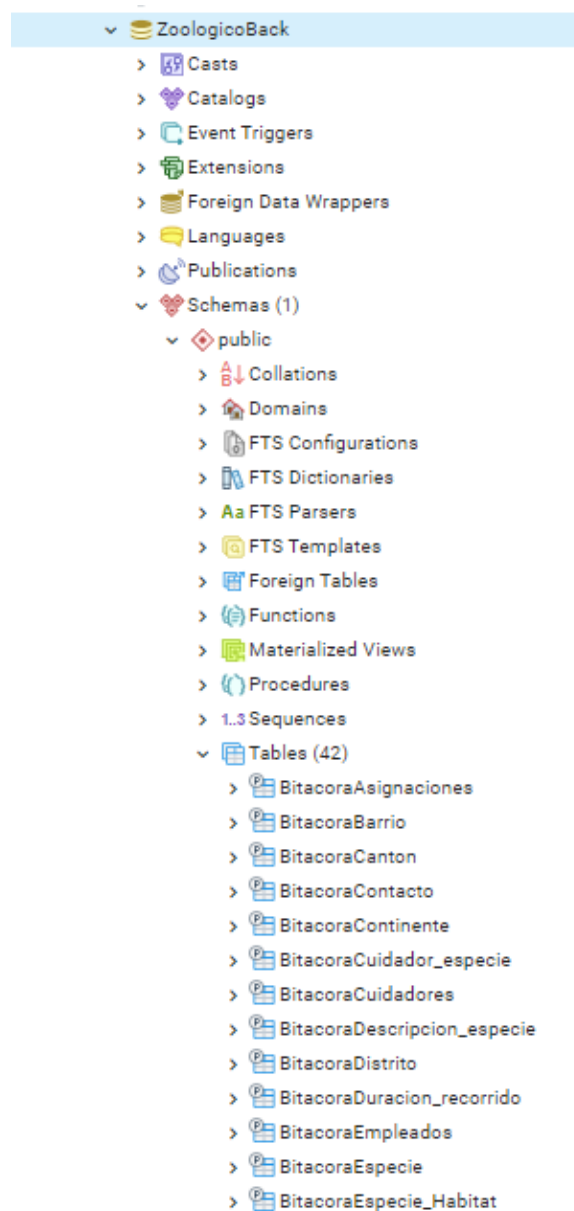
Para la conexión remota utilizamos PuTTY, nos aseguramos que la dirección IP sea correcta



2 Ingreso a PgAdmin

2.1 Ingreso a PgAdmin

Para este ejemplo, estaremos usando la base de datos llamada ZoologicoBack



2.2 Tablas en Base de datos

En la figura 1 se encuentran las Bitácoras. Además, las tablas correspondientes en la figura 2.

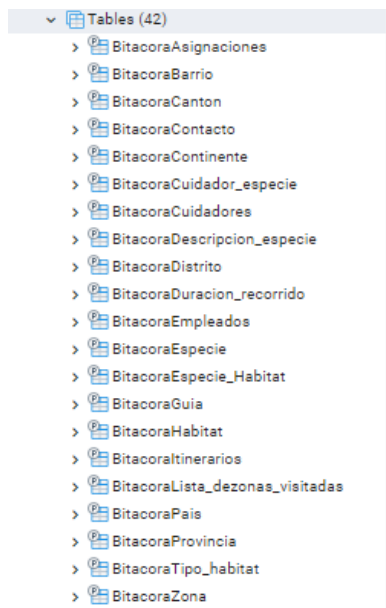


Figure 1: Bitácoras

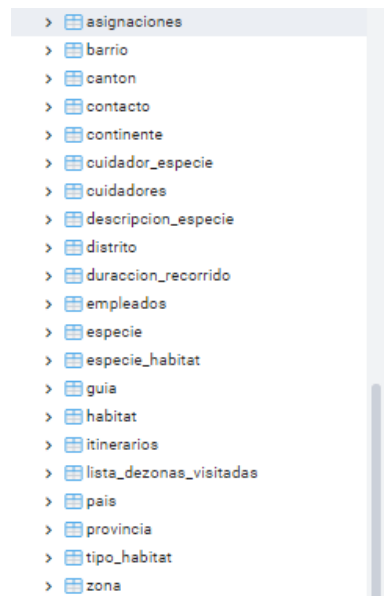


Figure 2: Tablas

2.3 Triggers en la base de datos

Mostrando los triggers de actualización en la base de datos

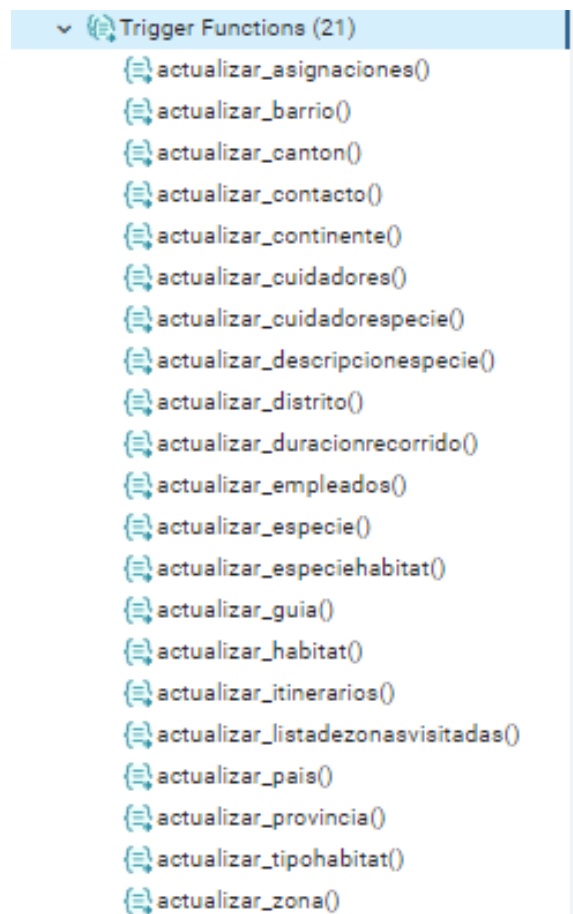


Figure 3: Triggers de actualización

3 Triggers Insertar

3.1 Tabla pais

3.1.1 Trigger que llama a la función que registra en la Tabla Bitácora País

```
CREATE FUNCTION ControlInsertarPais()  
returns TRIGGER  
LANGUAGE PLPGSQL  
AS $$  
  
DECLARE  
  
vusuario varchar(15);  
vid_pais integer;  
vnombre character(30);  
  
BEGIN  
vid_pais =NEW.id_pais;  
vnombre=NEW.nombre;  
  
SELECT user INTO vusuario;
```

Figure 4: Creación función trigger insertar pais

```
INSERT INTO public."BitacoraPais"  
  
VALUES (0,' ',vid_pais ,vnombre,vusuario,'I',now());  
  
RETURN NEW;  
  
END  
  
$$;  
  
CREATE TRIGGER Trigger_TablaPaisInsertar  
AFTER INSERT  
  
ON public.pais  
--CALL EL TRIGGER  
  
FOR EACH ROW EXECUTE PROCEDURE ControlInsertarPais();
```

Figure 5: Creación trigger

3.1.2 Inserción y materialización del Trigger en la bitácora país

```
insert into public.pais (id_pais, nombre) values ('507', 'Panama');  
insert into public.pais (id_pais, nombre) values ('52', 'Mexico');  
insert into public.pais(id_pais, nombre) values ('55', 'Brasil');  
insert into public.pais(id_pais, nombre) values ('7', 'Nicaragua');
```

Figure 6: Inserciones en tabla pais

```
SELECT id_pais, nombre  
FROM public.pais;
```

Figure 7: Sentencia *SELECT* para la tabla pais

id_pais [PK] integer	nombre character (30)
506	Costa Rica
1	Estados Unidos
507	Panama
52	Mexico
55	Brasil
7	Nicaragua

Figure 8: Entradas de la tabla *pais*

3.1.3 *SELECT BitacoraPais*

```
SELECT id_pais_old, nombre_old, id_pais_new, nombre_new, usuario, accion, "fechaRegistro"  
FROM public."BitacoraPais";
```

Figure 9: Seleccionando valores de la bitácora pais

Se puede observar las acciones realizadas en la imagen 10, junto con las respectivas acciones realizadas anteriormente, como actualizar, insertar.

id_pais_old integer	nombre_old character varying (30)	id_pais_new integer	nombre_new character varying (30)	usuario character varying (15)	accion character (1)	fechaRegistro date
1	USA	1	Estados Unidos	postgres	A	2021-06-23
0		507	Panama	postgres	I	2021-06-23
0		52	Mexico	postgres	I	2021-06-23
0		55	Brasil	postgres	I	2021-06-23
0		7	Nicaragua	postgres	I	2021-06-23

Figure 10: Entradas de bitácora pais

3.2 Tabla provincia

3.2.1 Trigger que llama a la función que registra en la Tabla Bitácora Provincia

```
CREATE FUNCTION ControlInsertarProvincia()  
returns TRIGGER  
LANGUAGE PLPGSQL  
AS $$  
  
DECLARE  
  
vusuario varchar(15);  
vid_provincia integer;  
vnombre character(30);  
vid_pais integer;  
BEGIN  
vid_provincia =NEW.id_provincia;  
vnombre=NEW.nombre;  
vid_pais =NEW.id_pais;  
SELECT user INTO vusuario;  
  
INSERT INTO public."BitacoraProvincia"
```

Figure 11: Creación función trigger insertar provincia

```
VALUES (0,'',0,vid_provincia ,vnombre,vid_pais,vusuario,'I',now());  
  
RETURN NEW;  
  
END  
  
$$;  
  
CREATE TRIGGER Trigger_TablaProvinciaInsertar  
AFTER INSERT  
  
ON public.provincia
```

Figure 12: Creación trigger

3.2.2 Inserción y materialización del Trigger en la bitácora provincia

```
insert into public.provincia (id_provincia, nombre,id_pais) values ('5', 'Guanacaste','506');  
insert into public.provincia (id_provincia, nombre,id_pais) values ('3', 'Cartago','506');  
insert into public.provincia (id_provincia, nombre,id_pais) values ('6', 'Puntarenas','506');  
  
insert into public.provincia (id_provincia, nombre,id_pais) values ('4', 'Heredia','506');
```

Figure 13: Inserciones en tabla provincia

```
SELECT id_provincia, nombre, id_pais  
FROM public.provincia;
```

Figure 14: Sentencia *SELECT* para la tabla provincia

id_provincia [PK] integer	nombre character (30)	id_pais integer
7	Guanacaste	506
2	Alajuela	506
8	Limon	506
1	Chepe	506
5	Guanacaste	506
3	Cartago	506
6	Puntarenas	506
4	Heredia	506

Figure 15: Entradas de la tabla *provincia*

3.2.3 *SELECT BitacoraProvincia*

```
SELECT id_provincia_old, nombre_old, id_pais_old, id_provincia_new, nombre_new, id_pais_new, usuario, accion, "fechaRegistro"
FROM public."BitacoraProvincia";
```

Figure 16: Seleccionando valores de la bitácora provincia

Se puede observar las acciones realizadas en la imagen 17, junto con las respectivas acciones realizadas anteriormente, como actualizar, insertar.

id_provincia_old integer	nombre_old character varying (30)	id_pais_old integer	id_provincia_new integer	nombre_new character varying (30)	id_pais_new integer	usuario character varying (15)	accion character (1)	fechaRegistro date
1	San Jose	506	1	Chepe	506	postgres	A	2021-06-23
0		0	5	Guanacaste	506	postgres	I	2021-06-23
0		0	3	Cartago	506	postgres	I	2021-06-23
0		0	6	Puntarenas	506	postgres	I	2021-06-23
0		0	4	Heredia	506	postgres	I	2021-06-23

Figure 17: Entradas de bitácora provincia

3.3 Tabla canton

3.3.1 Trigger que llama a la función que registra en la Tabla Bitácora Canton

```
CREATE FUNCTION ControlInsertarCanton()  
returns TRIGGER  
LANGUAGE PLPGSQL  
AS $$  
  
DECLARE  
  
vusuario varchar(15);  
vid_canton integer;  
vnombre character(30);  
vid_provincia integer;  
vid_pais integer;  
BEGIN  
vid_canton =NEW.id_canton;  
vnombre=NEW.nombre;  
vid_provincia =NEW.id_provincia;  
vid_pais=NEW.id_pais;  
SELECT user INTO vusuario;
```

Figure 18: Creación función trigger insertar canton

```
INSERT INTO public."BitacoraCanton"  
  
VALUES (0,' ',0,0,vid_canton ,vnombre,vid_provincia,vid_pais,vusuario,'I',now());  
  
RETURN NEW;  
  
END  
  
$$;  
  
CREATE TRIGGER Trigger_TablaCantonInsertar  
AFTER INSERT  
  
ON public.canton  
--CALL EL TRIGGER
```

Figure 19: Creación trigger

3.3.2 Inserción y materialización del Trigger en la bitácora canton

```
insert into public.canton (id_canton, nombre,id_provincia,id_pais) values ('2', 'Catie','2','1');
insert into public.canton (id_canton, nombre,id_provincia,id_pais) values ('3', 'Paxton','3','1');
insert into public.canton (id_canton, nombre,id_provincia,id_pais) values ('4', 'Liva','4','1');
insert into public.canton (id_canton, nombre,id_provincia,id_pais) values ('5', 'Ambrosius','5','1');
```

Figure 20: Inserciones en tabla canton

```
SELECT id_canton, nombre, id_provincia, id_pais
FROM public.canton;
```

Figure 21: Sentencia *SELECT* para la tabla canton

id_canton [PK] integer	nombre character (30)	id_provincia integer	id_pais integer
56	Tilaran	7	506
1	Sabana	1	506
2	Catie	2	1
3	Paxton	3	1
4	Liva	4	1
5	Ambrosius	5	1

Figure 22: Entradas de la tabla *canton*

3.3.3 *SELECT BitacoraCanton*

Se puede observar las acciones realizadas en la imagen 23, junto con las respectivas acciones realizadas anteriormente, como actualizar, insertar.

id_canton_old integer	nombre_old character varying (30)	id_provincia_old integer	id_pais_old integer	id_canton_new integer	nombre_new character varying (30)	id_provincia_new integer	id_pais_new integer	usuario character varying (15)
1	San Jose	1	506	1	Sabana	1	506	postgres
0		0	0	2	Catie	2	1	postgres
0		0	0	3	Paxton	3	1	postgres
0		0	0	4	Liva	4	1	postgres
0		0	0	5	Ambrosius	5	1	postgres

Figure 23: Entradas de bitácora canton

3.4 Tabla distrito

3.4.1 Trigger que llama a la función que registra en la Tabla Bitácora Distrito

```
CREATE FUNCTION ControlInsertarDistrito()  
returns TRIGGER  
LANGUAGE PLPGSQL  
AS $$  
  
DECLARE  
vid_pais integer;  
vusuario varchar(15);  
vid_distrito integer;  
vid_canton integer;  
vnombre character(30);  
vid_provincia integer;  
BEGIN  
vid_canton =NEW.id_canton;  
vnombre=NEW.nombre;  
vid_provincia =NEW.id_provincia;  
vid_distrito=NEW.id_distrito;
```

Figure 24: Creación función trigger insertar distrito

```
vid_distrito=NEW.id_distrito;  
vid_pais =NEW.id_pais;  
SELECT user INTO vusuario;  
  
INSERT INTO public."BitacoraDistrito"  
  
VALUES (0,' ',0,0,0,vid_distrito,vnombre,vid_canton ,vid_provincia,vid_pais,vusuario,'I',now());  
  
RETURN NEW;  
  
END  
  
$$;  
  
CREATE TRIGGER Trigger_TablaDistritoInsertar  
AFTER INSERT  
  
ON public.distrito  
--CALL EL TRIGGER
```

Figure 25: Creación trigger

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

Figure 26: Creación trigger

3.4.2 Inserción y materialización del Trigger en la bitácora distrito

```
insert into public.distrito (id_distrito, nombre,id_canton,id_provincia,id_pais) values ('1', 'San Roque','1','1','506');
insert into public.distrito (id_distrito, nombre,id_canton,id_provincia,id_pais) values ('2', 'Cañas','2','2','506');

insert into public.distrito (id_distrito, nombre,id_canton,id_provincia,id_pais) values ('5', 'Ambrosius','5','5','506');
```

Figure 27: Inserciones en tabla distrito

```
SELECT id_distrito, nombre, id_canton, id_provincia, id_pais
FROM public.distrito;
```

Figure 28: Sentencia *SELECT* para la tabla distrito

id_distrito [PK] integer	nombre character (30)	id_canton integer	id_provincia integer	id_pais integer
100	Quebrada Grande ...	56	7	506
101	Tronadora	56	7	506
102	Tilaran	56	7	506
1	San Roque	1	1	506
2	Cañas	2	2	506
5	Ambrosius	5	5	506

Figure 29: Entradas de la tabla *distrito*

3.4.3 *SELECT BitacoraDistrito*

Se puede observar las acciones realizadas en la imagen 30, junto con las respectivas acciones realizadas anteriormente, como actualizar, insertar.

id_distrito_old integer	nombre_old character varying (30)	id_canton_old integer	id_provincia_old integer	id_pais_old integer	id_distrito_new integer	nombre_new character varying (30)	id_canton_new integer	id_provincia_new integer
0		0	0	0	1	San Roque	1	1
0		0	0	0	2	Cañas	2	2
0		0	0	0	5	Ambrosius	5	5

Figure 30: Entradas de bitácora distrito

3.5 Tabla barrio

3.5.1 Trigger que llama a la función que registra en la Tabla Bitácora Barrio

```
CREATE FUNCTION ControlInsertarBarrio()
returns TRIGGER
LANGUAGE PLPGSQL
AS $$

DECLARE

vusuario varchar(15);
vid_distrito integer;
vid_canton integer;
vnombre character(30);
vid_provincia integer;
vid_barrio integer;
vid_pais integer;
BEGIN
vid_canton =NEW.id_canton;
vnombre=NEW.nombre;
vid_provincia =NEW.id_provincia;
```

Figure 31: Creación función trigger insertar barrio

```
vid_provincia =NEW.id_provincia;
vid_distrito=NEW.id_distrito;
vid_barrio=NEW.id_barrio;
vid_pais=NEW.id_pais;
SELECT user INTO vusuario;

INSERT INTO public."BitacoraBarrio"

VALUES (0,' ',0,0,0,0,vid_barrio,vnombre,vid_canton ,vid_distrito,vid_provincia,vid_pais,vusuario,'I',now());

RETURN NEW;

END

$$;

CREATE TRIGGER trigger_TablaBarrioInsertar
AFTER INSERT
```

Figure 32: Creación trigger

```
ON public.barrio
--CALL EL TRIGGER

FOR EACH ROW EXECUTE PROCEDURE ControlInsertarBarrio();

--fin del trigger de insercion en la tabla distrito
```

Figure 33: Creación trigger

3.5.2 Inserción y materialización del Trigger en la bitácora barrio

```
SELECT id_barrio, nombre, id_distrito, id_canton, id_provincia, id_pais
FROM public.barrio;
```

Figure 34: Inserciones en tabla barrio

```
SELECT id_barrio, nombre, id_distrito, id_canton, id_provincia, id_pais
FROM public.barrio;
```

Figure 35: Sentencia *SELECT* para la tabla barrio

id_barrio [PK] integer	nombre character (30)	id_distrito integer	id_canton integer	id_provincia integer	id_pais integer
501	Bonanza	--	102	56	7
503	San Isais	--	101	56	7
500	BIEN PASO	--	100	56	7
1	San Roque	--	1	1	506
2	Cuñas	--	2	2	506

Figure 36: Entradas de la tabla *barrio*

3.5.3 SELECT BitacoraBarrio

```

701 SELECT id_barrio_old, nombre_old, id_distrito_old, id_canton_old, id_provincia_old,
702        id_pais_old, id_barrio_new, nombre_new, id_distrito_new, id_canton_new,
703        id_provincia_new, id_pais_new, usuario, accion, "fechaRegistro"
704        FROM public."BitacoraBarrio";
705

```

Figure 37: Seleccionando valores de la bitácora barrio

Se puede observar las acciones realizadas en la imagen 38, junto con las respectivas acciones realizadas anteriormente, como actualizar, insertar.

id_barrio_old integer	nombre_old character varying (30)	id_distrito_old integer	id_canton_old integer	id_provincia_old integer	id_pais_old integer	id_barrio_new integer	nombre_new character varying (30)	id_distrito_new integer	id_canton_new integer
500	La plata	100	56	7	506	500	BUEN PASO	100	56
0		0	0	0	0	1	San Roque	1	1
0		0	0	0	0	2	Cañas	2	2

Figure 38: Entradas de bitácora barrio

id_canton_new integer	id_provincia_new integer	id_pais_new integer	usuario character varying (15)	accion character (1)	fechaRegistro date
56	7	506	postgres	A	2021-06-23
1	1	506	postgres	I	2021-06-23
2	2	506	postgres	I	2021-06-23

Figure 39: Entradas de bitácora barrio

3.6 Tabla empleados

3.6.1 Trigger que llama a la función que registra en la Tabla Bitácora Empleados

```
CREATE FUNCTION ControlInsertarEmpleado()  
returns TRIGGER  
LANGUAGE PLPGSQL  
AS $$  
  
DECLARE  
  
vid_empleado integer;  
vprimer_nombre character(30);  
vsegundo_nombre character(30);  
vprimer_apellido character(30);  
vsegundo_apellido character(30);  
vusuario varchar(15);  
vid_distrito integer;  
vid_canton integer;  
  
vid_provincia integer;  
vid_barrio integer;
```

Figure 40: Creación función trigger insertar empleados

```
vid_barrio integer;  
vid_pais integer;  
BEGIN  
vid_empleado=NEW.id_empleado;  
vprimer_nombre=NEW.primer_nombre;  
vsegundo_nombre =NEW.segundo_nombre;  
vprimer_apellido =NEW.primer_apellido;  
vsegundo_apellido=NEW.segundo_apellido;  
vid_canton =NEW.id_canton;  
  
vid_provincia =NEW.id_provincia;  
vid_distrito=NEW.id_distrito;  
vid_barrio=NEW.id_barrio;  
vid_pais=NEW.id_pais;  
SELECT user INTO vusuario;  
  
INSERT INTO public."BitacoraEmpleados"
```

Figure 41: Creación trigger

```
VALUES (0, ' ', ' ', ' ', ' ', ' ', 0,0,0,0,0,vid_empleado,vprimer_nombre,vsegundo_nombre,vprimer_apellido,'
```

Figure 42: Creación trigger

```
,vsegundo_apellido,vid_provincia,vid_canton ,vid_distrito,vid_barrio,vid_pais,vusuario,'I',now());
```

Figure 43: Creación trigger

```
RETURN NEW;
```

```
END
```

```
$$;
```

```
CREATE TRIGGER Trigger_TablaEmpleadoInsertar  
AFTER INSERT
```

```
ON public.empleados
```

```
--CALL EL TRIGGER
```

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

Figure 44: Creación trigger

3.6.2 Inserción y materialización del Trigger en la bitácora empleados

```
insert into public.empleados(  
insert into public.empleados  
  
insert into public.empleados(  
insert into public.empleados  
  
insert into public.empleados(  
insert into public.empleados  
  
insert into public.empleados(  
insert into public.empleados
```

Figure 45: Inserciones en tabla empleados

```
id_empleado,primer_nombre,segundo_nombre,primer_apellido,segundo_apellido,id_provincia, id_canton,id_distrito,id_barrio,id_pais)  
(id_empleado,primer_nombre,segundo_nombre,primer_apellido,segundo_apellido,id_provincia, id_canton,id_distrito,id_barrio,id_pais)  
id_empleado,primer_nombre,segundo_nombre,primer_apellido,segundo_apellido,id_provincia, id_canton,id_distrito,id_barrio,id_pais)  
(id_empleado,primer_nombre,segundo_nombre,primer_apellido,segundo_apellido,id_provincia, id_canton,id_distrito,id_barrio,id_pais)  
id_empleado,primer_nombre,segundo_nombre,primer_apellido,segundo_apellido,id_provincia, id_canton,id_distrito,id_barrio,id_pais)  
(id_empleado,primer_nombre,segundo_nombre,primer_apellido,segundo_apellido,id_provincia, id_canton,id_distrito,id_barrio,id_pais)  
id_empleado,primer_nombre,segundo_nombre,primer_apellido,segundo_apellido,id_provincia, id_canton,id_distrito,id_barrio,id_pais)  
(id_empleado,primer_nombre,segundo_nombre,primer_apellido,segundo_apellido,id_provincia, id_canton,id_distrito,id_barrio,id_pais)  
id_empleado,primer_nombre,segundo_nombre,primer_apellido,segundo_apellido,id_provincia, id_canton,id_distrito,id_barrio,id_pais)
```

Figure 46: Inserciones en tabla empleados

```

values ('1234','Julian','Daniel','ordonez','Guzman','1','1','1','1','1');
i) values ('2345', 'Annie', 'Astrix', 'Jammet', 'Gricks','2','2','2','501','1');

values ('5001','waldo','Daniel','ordonez','Guzman','1','1','1','1','1');
i) values ('2001', 'Annie', 'Astrix', 'Jammet', 'Gricks','2','2','2','501','1');

values ('5003','oden','Daniel','ordonez','Guzman','1','1','1','1','1');
i) values ('2004', 'Michea', 'Astrix', 'Jammet', 'Gricks','2','2','2','501','1');

values ('5010','luffy','Daniel','hernandez','Guzman','1','1','1','1','1');
i) values ('2040', 'zoro', 'Astrix', 'Jammet', 'Gricks','2','2','2','501','1');

values ('5030','matsuda','Daniel','garcia','Guzman','1','1','1','1','1');

```

Figure 47: Inserciones en tabla empleados

```

SELECT id_employe, primer_nombre, segundo_nombre, primer_apellido, segundo_apellido, id_pais, id_estado, id_ciudad, id_oficina, id_departamento, id_pais;
FROM public.employees;

```

Figure 48: Sentencia *SELECT* para la tabla empleados

ID_employe	primer_nombre	segundo_nombre	primer_apellido	segundo_apellido	ID_pais	ID_estado	ID_ciudad	ID_oficina	ID_departamento
1234	Julian	Daniel	ordonez	Guzman	1	1	1	1	1
2345	Annie	Astrix	Jammet	Gricks	2	2	2	501	1
5001	waldo	Daniel	ordonez	Guzman	1	1	1	1	1
2001	Annie	Astrix	Jammet	Gricks	2	2	2	501	1
5003	oden	Daniel	ordonez	Guzman	1	1	1	1	1
2004	Michea	Astrix	Jammet	Gricks	2	2	2	501	1
5010	luffy	Daniel	hernandez	Guzman	1	1	1	1	1
2040	zoro	Astrix	Jammet	Gricks	2	2	2	501	1
5030	matsuda	Daniel	garcia	Guzman	1	1	1	1	1

Figure 49: Entradas de la tabla *empleados*

3.6.3 SELECT BitacoraEmpleados

```
SELECT id_empleado_old, primer_nombre_old, segundo_nombre_old, primer_apellido_old,
segundo_apellido_old, id_provincia_old, id_canton_old, id_distrito_old,
id_barrio_old, id_pais_old, id_empleado_new, primer_nombre_new, segundo_nombre_new,
primer_apellido_new, segundo_apellido_new, id_provincia_new, id_canton_new,
id_distrito_new, id_barrio_new, id_pais_new, usuario, accion, "fechaRegistro"

FROM public."BitacoraEmpleados";
```

Figure 50: Seleccionando valores de la bitácora empleados

Se puede observar las acciones realizadas en la imagen 52, junto con las respectivas acciones realizadas anteriormente, como actualizar, insertar.

id_empleado_new integer	primer_nombre_new character varying (30)	segundo_nombre_new character varying (30)	primer_apellido_new character varying (30)	segundo_apellido_new character varying (30)	id_provincia_new integer	id_canton_new integer	id_distrito_new integer	id_barrio_new integer
2077	dorian	[null]	sjdfbsjbsdf	fdtsdf	7	56	100	500
2077	dorian	[null]	Rodriguez	fdtsdf	7	56	100	500
2077	dorian	[null]	Rodriguez	fdtsdf	7	56	100	500
2077	dorian	[null]	fghfghfghfg	fdtsdf	7	56	100	500
1234	Julian	Daniel	ordonez	Guzman	1	1	1	1
2345	Annie	Astrix	Jammet	Gricks	2	2	2	501
5001	waldo	Daniel	ordonez	Guzman	1	1	1	1
2001	Annie	Astrix	Jammet	Gricks	2	2	2	501
5003	oden	Daniel	ordonez	Guzman	1	1	1	1
2004	Mitchae	Astrix	Jammet	Gricks	2	2	2	501
5010	luffy	Daniel	hernandez	Guzman	1	1	1	1
2040	zoro	Astrix	Jammet	Gricks	2	2	2	501
5030	matsuda	Daniel	garcla	Guzman	1	1	1	1

Figure 51: Entradas de bitácora empleados

id_pais_new integer	usuario character varying (15)	accion character (1)	fechaRegistro date
506	postgres	A	2021-06-23
506	postgres	A	2021-06-23
506	postgres	A	2021-06-23
506	postgres	A	2021-06-23
1	postgres	I	2021-06-23
1	postgres	I	2021-06-23
1	postgres	I	2021-06-23
1	postgres	I	2021-06-23
1	postgres	I	2021-06-23
1	postgres	I	2021-06-23
1	postgres	I	2021-06-23
1	postgres	I	2021-06-23
1	postgres	I	2021-06-23
1	postgres	I	2021-06-23

Figure 52: Entradas de bitácora empleados

3.7 Tabla contacto

3.7.1 Trigger que llama a la función que registra en la Tabla Bitácora Contacto

```
CREATE FUNCTION ControlInsertarContacto()  
returns TRIGGER  
LANGUAGE PLPGSQL  
AS $$  
  
DECLARE  
  
vusuario varchar(15);  
vid_contacto integer;  
vid_empleado integer;  
vnumero_de_telefono character(30);  
vnumero_secundario character(30);  
vcorreo_electronico character(60);  
vcorreo_electronico_secundario character(60);  
BEGIN  
vid_contacto =NEW.id_contacto;  
vid_empleado =NEW.id_empleado;
```

Figure 53: Creación función trigger insertar contactos

```
vnumero_de_telefono=NEW.numero_de_telefono;  
vnumero_secundario =NEW.numero_secundario;  
vcorreo_electronico=NEW.correo_electronico;  
vcorreo_electronico_secundario=NEW.correo_electronico_secundario;  
SELECT user INTO vusuario;  
  
INSERT INTO public."BitacoraContacto"
```

Figure 54: Creación trigger

3.7.2 Inserción y materialización del Trigger en la bitácora contactos

```
insert into contacto (id_contacto, id_empleado, numero_de_telefono, numero_secundario, correo_electronico, i
insert into contacto (id_contacto, id_empleado, numero_de_telefono, numero_secundario, correo_electronico, i

insert into contacto (id_contacto, id_empleado, numero_de_telefono, numero_secundario, correo_electronico, i
insert into contacto (id_contacto, id_empleado, numero_de_telefono, numero_secundario, correo_electronico, i
```

Figure 55: Inserciones en tabla contactos

```
correo_electronico_secundario)
correo_electronico_secundario)

correo_electronico_secundario)
correo_electronico_secundario)
```

Figure 56: Inserciones en tabla contactos

```
values (4, 5001, '401 964 5175', '564-568-9916', 'knibley3@bloomberg.com', 'hziemen3@ebay.co.uk');
values (5, 2004, '250 223 5071', '149-554-0359', 'prubury4@china.com.cn', 'syushankin4@hc360.com');

values (10, 5003, '221 346 4883', '494-934-3852', 'tduckham1@ustream.tv', 'bshuttlewood1@mayoclinic.com');
values (15, 2001, '131 387 4436', '492-215-0061', 'jdowzell2@newyorker.com', 'cpehrsson2@amazon.co.uk');
```

Figure 57: Inserciones en tabla contactos

id_contacto [PK] integer	id_empleado integer	numero_de_telefono character (30)	numero_secundario character (30)	correo_electronico character (60)	correo_electronico_secundario character (60)
1	5622	8545621	861144	sdfdd@mail.com	sdfsd@mail2.com
2	2077	5564555	815555	sdfdd@mail.com	sdfsd@mail2.com
3	7882	1554564565465	85222	sdfdd@mail.com	sdfsd@mail2.com
4	5001	401 964 5175	564-568-9916	knibley3@bloomberg.com	hziemen3@ebay.co.uk
5	2004	250 223 5071	149-554-0359	prubury4@china.com.cn	syushankin4@hc360.com
10	5003	221 346 4883	494-934-3852	tduckham1@ustream.tv	bshuttlewood1@mayoclinic.com
15	2001	131 387 4436	492-215-0061	jdowzell2@newyorker.com	cpehrsson2@amazon.co.uk

Figure 58: Inserciones en tabla contactos

3.7.3 SELECT BitacoraContacto

```
SELECT id_contacto_old, id_empleado_old, numero_de_telefono_old, numero_secundario_old,
       correo_electronico_old, correo_electronico_secundario_old, id_contacto_new,
       id_empleado_new, numero_de_telefono_new, numero_secundario_new,
       correo_electronico_new, correo_electronico_secundario_new,
       usuario, accion, "fechaRegistro"
FROM public."BitacoraContacto";
```

Figure 59: Seleccionando valores de la bitácora contactos

Se puede observar las acciones realizadas en la imagen 60, junto con las respectivas acciones realizadas anteriormente, como actualizar, insertar.

id_contacto_new integer	id_empleado_new integer	numero_de_telefono_new character (30)	numero_secundario_new character varying (60)	correo_electronico_new character varying (60)	correo_electronico_secundario_new character (60)	usuario character varying (
3	7882	1554564565465	85222	sdfsdf@mail2.com	sdfsdf@mail2.com	postgres
4	5001	401 964 5175	564-568-9916	knibley3@bloomberg.com	hziemen3@ebay.co.uk	postgres
5	2004	250 223 5071	149-554-0359	prubury4@china.com.cn	syushankin4@hc360.com	postgres
10	5003	221 346 4883	494-934-3852	tduckham1@ustream.tv	bshuttlewood1@mayoclinic.com	postgres
15	2001	131 387 4436	492-215-0061	jdowzell2@newyorker.com	cpehrsson2@amazon.co.uk	postgres

Figure 60: Entradas de bitácora contacto

accion character (1)	fechaRegistro date
A	2021-06-23
I	2021-06-23
I	2021-06-23
I	2021-06-23
I	2021-06-23

Figure 61: Entradas de bitácora contacto

3.8 Tabla guía

3.8.1 Trigger que llama a la función que registra en la Tabla Bitácora Guía

```
CREATE FUNCTION ControlInsertarGuia()  
returns TRIGGER  
LANGUAGE PLPGSQL  
AS $$  
  
DECLARE  
  
vusuario varchar(15);  
vid_guia integer;  
vid_empleado integer;  
vfecha_deingreso DATE;  
  
BEGIN  
vid_guia =NEW.id_guia;  
vid_empleado =NEW.id_empleado;  
vfecha_deingreso=NEW.fecha_deingreso laboral;
```

Figure 62: Creación función trigger insertar guia

```
BEGIN  
vid_guia =NEW.id_guia;  
vid_empleado =NEW.id_empleado;  
vfecha_deingreso=NEW.fecha_deingreso_laboral;  
  
SELECT user INTO vusuario;  
  
INSERT INTO public."BitacoraGuia"  
  
VALUES (0,0,null,vid_guia,vid_empleado,vfecha_deingreso,vusuario,'I',now());  
  
RETURN NEW;
```

Figure 63: Creación trigger

```
|  
CREATE TRIGGER Trigger_TablaGuiaInsertar  
AFTER INSERT  
  
ON public.guia  
--CALL EL TRIGGER  
  
FOR EACH ROW EXECUTE PROCEDURE ControlInsertarGuia();
```

Figure 64: Creación trigger

3.8.2 Inserción y materialización del Trigger en la bitácora guia

```
insert into guia (id_guia, id_empleado, fecha_deingreso_laboral) values (1, 5030, '2020-07-09');
insert into guia (id_guia, id_empleado, fecha_deingreso_laboral) values (2, 2001, '2020-08-19');
insert into guia (id_guia, id_empleado, fecha_deingreso_laboral) values (4, 5001, '2021-02-04');
```

Figure 65: Inserciones en tabla guia

	id_guia [PK] integer	id_empleado integer	fecha_deingreso_laboral date
1	900	2077	2005-07-23
2	1	5030	2020-07-09
3	2	2001	2020-08-19
4	4	5001	2021-02-04

Figure 66: Entradas en la tabla guia

3.8.3 *SELECT BitacoraGuia*

```
SELECT id_guia_old, id_empleado_old, fecha_deingreso_laboral_old, id_guia_new,  
       id_empleado_new, fecha_deingreso_laboral_new, usuario, accion, "fechaRegistro"  
  
FROM public."BitacoraGuia";
```

Figure 67: Seleccionando valores de la bitácora guía

Se puede observar las acciones realizadas en la imagen 68, junto con las respectivas acciones realizadas anteriormente, como actualizar, insertar.

id_guia_new integer	id_empleado_new integer	fecha_deingreso_laboral_new date	usuario character varying (15)	accion character (1)	fechaRegistro date
900	2077	2005-07-23	postgres	A	2021-06-23
1	5030	2020-07-09	postgres	I	2021-06-23
2	2001	2020-08-19	postgres	I	2021-06-23
4	5001	2021-02-04	postgres	I	2021-06-23

Figure 68: Entradas de bitácora guía

3.9 Tabla duracion_recorrido

3.9.1 Trigger que llama a la función que registra en la Tabla Bitácora Duracion_recorrido

```
CREATE FUNCTION ControlInsertarDuracionRecorrido()  
returns TRIGGER  
LANGUAGE PLPGSQL  
AS $$  
  
DECLARE  
  
vusuario varchar(15);  
vid_duracion_recorrido integer;  
vhora integer;  
vminutos integer;  
vsegundos integer;
```

Figure 69: Creación función trigger insertar duracion de recorrido

```
BEGIN  
vid_duracion_recorrido =NEW.id_duracion_recorrido;  
vhora =NEW.hora;  
vminutos=NEW.minutos;  
vsegundos=NEW.segundos;  
  
SELECT user INTO vusuario;  
  
INSERT INTO public."BitacoraDuracion_recorrido"  
  
VALUES (0,0,0,0,vid_duracion_recorrido,vhora,vminutos,vsegundos,vusuario,'I',now());  
  
RETURN NEW;  
  
END
```

Figure 70: Creación trigger

```
CREATE TRIGGER Trigger_TablaDuracion_Recorrido_Insertar
AFTER INSERT

ON public.duracion_recorrido
--CALL EL TRIGGER

FOR EACH ROW EXECUTE PROCEDURE ControlInsertarDuracionRecorrido();

--Trigger eliminacion de duracion recorrido
```

Figure 71: Creación trigger

3.9.2 Inserción y materialización del Trigger en la bitácora duracion recorrido

```
insert into duracion_recorrido (id_duracion_recorrido, hora, minutos, segundos) values (1, '11', '01 ', '00');  
insert into duracion_recorrido (id_duracion_recorrido, hora, minutos, segundos) values (2, '5', '14 ', '00 ');  
insert into duracion_recorrido(id_duracion_recorrido, hora, minutos, segundos) values (3, '5', '47 ', '00');
```

Figure 72: Inserciones en tabla duracion_recorrido

3.9.3 *SELECT BitacoraDuracionRecorrido*

```
SELECT id_duracion_recorrido_old, hora_old, minutos_old, segundos_old,  
       id_duracion_recorrido_new, hora_new, minutos_new, segundos_new,  
       usuario, accion, "fechaRegistro"  
  
FROM public."BitacoraDuracion_recorrido";
```

Figure 73: Seleccionando valores de la bitácora duracion recorrido

Se puede observar las acciones realizadas en la imagen 74, junto con las respectivas acciones realizadas anteriormente, como actualizar, insertar.

id_duracion_recorrido_new integer	hora_new integer	minutos_new integer	segundos_new integer	usuario character varying (15)	accion character (1)	fechaRegistro date
90	1	30	0	postgres	A	2021-06-23
1	11	1	0	postgres	I	2021-06-23
2	5	14	0	postgres	I	2021-06-23
3	5	47	0	postgres	I	2021-06-23

Figure 74: Entradas de bitácora duracion recorrido

3.10 Tabla itinerarios

3.10.1 Trigger que llama a la función que registra en la Tabla Bitácora Itinerarios

```
CREATE FUNCTION ControlInsertarItinerarios()
returns TRIGGER
LANGUAGE PLPGSQL
AS $$

DECLARE

vusuario varchar(15);
vcodigo varchar(30);
vid_duracion_recorrido integer;
vlongitud_recorrido double precision;
vnumero_maximodevisitante_autorizados integer;
vnumero_dedistintasespecies_quevisita integer;

vsegundos integer;
```

Figure 75: Creación función trigger insertar itinerario

```
BEGIN
vcodigo=NEW.codigo;

vid_duracion_recorrido =NEW.id_duracion_recorrido;
vlongitud_recorrido =NEW.longitud_recorrido;
vnumero_maximodevisitante_autorizados=NEW.numero_maximodevisitante_autorizados;
vnumero_dedistintasespecies_quevisita =NEW.numero_dedistintasespecies_quevisita ;

SELECT user INTO vusuario;

INSERT INTO public."BitacoraItinerarios"
```

Figure 76: Creación trigger

```
VALUES (0,0,0,0,0,vcodigo,vid_duracion_recorrido,vlongitud_recorrido,vnumero_maximodevisitante_autorizados,'
```

Figure 77: Creación trigger

```
,vnumero_dedistintasespecies_quevisita,vusuario,'I',now()));
```

Figure 78: Creación trigger

```
RETURN NEW;  
  
END  
  
$$;  
  
CREATE TRIGGER Trigger_TablaItinerarios_Insertar  
AFTER INSERT  
  
ON public.itinerarios  
--CALL EL TRIGGER  
  
FOR EACH ROW EXECUTE PROCEDURE ControlInsertarItinerarios();
```

Figure 79: Creación trigger

3.10.2 Inserción y materialización del Trigger en la bitácora itinerario

```
insert into itinerarios (codigo, id_duracion_recorrido, longitud_recorrido, numero_maximodevisitante_autorizados,  
insert into itinerarios (codigo, id_duracion_recorrido, longitud_recorrido, numero_maximodevisitante_autorizados,  
  
insert into itinerarios (codigo, id_duracion_recorrido, longitud_recorrido, numero_maximodevisitante_autorizados,  
insert into itinerarios (codigo, id_duracion_recorrido, longitud_recorrido, numero_maximodevisitante_autorizados,
```

Figure 80: Inserciones en tabla itinerarios

```
numero_dedistintasespecies_quevisita) values ('GH22', 1, 30.50, 30, 10);  
numero_dedistintasespecies_quevisita) values ('BH26', 2, 20.3, 20, 25);  
  
numero_dedistintasespecies_quevisita) values ('GH30', 1, 300.4, 33, 32);  
numero_dedistintasespecies_quevisita) values ('GH45', 2, 400.5, 44, 43);
```

. . .

Figure 81: Inserciones en tabla itinerarios

3.10.3 *SELECT BitacoraItinerarios*

```
SELECT codigo_old, id_duracion_recorrido_old, longitud_recorrido_old,
       numero_maximodevisitante_autorizados_old, numero_dedistintasespecies_quevisita_old,
       codigo_new, id_duracion_recorrido_new, longitud_recorrido_new,
       numero_maximodevisitante_autorizados_new, numero_dedistintasespecies_quevisita_new,
       usuario, accion, "fechaRegistro"
FROM public."BitacoraItinerarios";
```

Figure 82: Seleccionando valores de la bitácora itinerarios

Se puede observar las acciones realizadas en la imagen 83, junto con las respectivas acciones realizadas anteriormente, como actualizar, insertar.

codigo_new character varying (30)	id_duracion_recorrido_new integer	longitud_recorrido_new double precision	numero_maximodevisitante_autorizados_new integer	numero_dedistintasespecies_quevisita_new integer
3000	90	80.1	10	12
GH22	1	30.5	30	10
BH26	2	20.3	20	25
GH30	1	300.4	33	32
GH45	2	400.5	44	43

Figure 83: Entradas de bitácora itinerarios

3.11 Tabla asignaciones

3.11.1 Trigger que llama a la función que registra en la Tabla Bitácora Asignaciones

```
CREATE FUNCTION ControlInsertarAsignaciones()  
returns TRIGGER  
LANGUAGE PLPGSQL  
AS $$  
  
DECLARE  
  
vusuario varchar(15);  
vid_asignaciones integer;  
vid_guia integer;  
  
vhora_comienzo TIME;  
vcodigo varchar(30);  
  
BEGIN  
vid_asignaciones =NEW.id_asignaciones ;  
vid_guia =NEW.id_guia;  
vhora_comienzo =NEW.hora_comienzo;  
vcodigo =NEW.codigo ;
```

Figure 84: Creación función trigger insertar asignacion

```
VALUES (0,null,0,'',vid_asignaciones,vhora_comienzo ,vid_guia,vcodigo,vusuario,'I',now());  
  
RETURN NEW;  
  
END  
  
$$;  
  
CREATE TRIGGER Trigger_TablaInsertarAsignaciones  
AFTER INSERT  
  
ON public.asignaciones  
--CALL EL TRIGGER
```

Figure 85: Creación trigger

3.11.2 Inserción y materialización del Trigger en la bitácora asignaciones

```
insert into asignaciones (id_asignaciones, hora_comienzo, id_guia, codigo) values (4, '11:23:02', 1, 'GH22');  
insert into asignaciones (id_asignaciones, hora_comienzo, id_guia, codigo) values (5, '10:52:01', 2, 'BH26');  
insert into asignaciones (id_asignaciones, hora_comienzo, id_guia, codigo) values (10, '10:50:01', 2, 'BH26');
```

Figure 86: Inserciones en tabla asignaciones

3.11.3 *SELECT BitacoraAsignaciones*

```
SELECT id_asignaciones_old, hora_comienzo_old, id_guia_old, codigo_old,  
       id_asignaciones_new, hora_comienzo_new, id_guia_new, codigo_new,  
       usuario, accion, "fechaRegistro"  
FROM public."BitacoraAsignaciones";
```

Figure 87: Seleccionando valores de la bitácora asignaciones

Se puede observar las acciones realizadas en la imagen 88, junto con las respectivas acciones realizadas anteriormente, como actualizar, insertar.

id_asignaciones_new integer	hora_comienzo_new time without time zone	id_guia_new integer	codigo_new character varying (30)	usuario character varying (15)	accion character (1)	fechaRegistro date
1	07:30:00	900	3000	postgres	A	2021-06-23
4	11:23:02	1	GH22	postgres	I	2021-06-23
5	10:52:01	2	BH26	postgres	I	2021-06-23
10	10:50:01	2	BH26	postgres	I	2021-06-23

Figure 88: Entradas de bitácora asignaciones

3.12 Tabla continente

3.12.1 Trigger que llama a la función que registra en la Tabla Bitácora Continente

```
26
27 -- insertar continente
28 CREATE FUNCTION insertar_continente() RETURNS TRIGGER AS
29 $$
30 DECLARE
31
32     accion character(1) := 'I';
33     usuario varchar(30) := (select user);
34     fecha_registro date := now();
35     id_nulo integer := 0;
36     nombre_nulo character(1) := ' ';
37
38 BEGIN
39
40     INSERT INTO public."BitacoraContinente"
41     VALUES (id_nulo, nombre_nulo,
42
43             NEW.id_continente, NEW.nombre,
44
45             usuario, accion, fecha_registro);
46
47     RETURN NEW;
48
49 END
50 $$
51 LANGUAGE PLPGSQL;
```

Figure 89: Creación función trigger insertar continente

```
607
608 -- Trigger Insertar Continente
609 CREATE TRIGGER Trigger_InsertarContinente AFTER INSERT ON continente
610 FOR EACH ROW EXECUTE PROCEDURE insertar_continente();
611
612
```

Figure 90: Creación trigger

3.12.2 Inserción y materialización del Trigger en la bitácora continente

```
706 select * from continente;
707 INSERT INTO continente (id_continente, nombre) values(6, 'Antartica');
708 select * from "BitacoraContinente";
709
```

Notifications				Data Output				Messages				Explain			
				<div>id_continente</div> <div>[PK] integer</div>				<div>nombre</div> <div>character varying (30)</div>							
1				255				Europa							

3.12.3 *SELECT BitacoraContinente*

```
707
708 select * from continente;
709 INSERT INTO continente (id_continente, nombre) values(6, 'Antartica');
710 select * from "BitacoraContinente";
711
```

	id_continente_old integer	nombre_old character varying (30)	id_continente_new integer	nombre_new character varying (30)	usuario character varying (15)	accion character (1)	fechaRegistro date
1	255	America	255	Europa	postgres	A	2021-06-23
2	0		2	Antartica	postgres	I	2021-06-25
3	2	Antartica	0		postgres	D	2021-06-25
4	0		6	Antartica	postgres	I	2021-06-25

Figure 92: Seleccionando valores de la bitácora continente

3.13 Tabla cuidadores

3.13.1 Trigger que llama a la función que registra en la Tabla Bitácora Cuidadores

```
202 -- insertar cuidador
203 CREATE FUNCTION insertar_cuidador() RETURNS TRIGGER AS
204 $$
205 DECLARE
206     accion character(1) := 'I';
207     usuario varchar(30) := (select user);
208     fecha_registro date := now();
209     id_nulo integer := 0;
210     fecha_de_ingreso_alparque_nulo date := '-infinity';
211
212 BEGIN
213
214     INSERT INTO public."BitacoraCuidadores"
215     VALUES (id_nulo, fecha_de_ingreso_alparque_nulo,
216
217             NEW.id_cuidador, NEW.fecha_ingreso_alparque,
218
219             usuario, accion, fecha_registro);
220
221 RETURN NEW;
222
223 END
224 $$
225 LANGUAGE PLPGSQL;
```

Figure 93: Creación función trigger insertar cuidador

```
639 |
640 -- Trigger Insertar Cuidador
641 CREATE TRIGGER Trigger_InsertarCuidador AFTER INSERT ON cuidadores
642 FOR EACH ROW EXECUTE PROCEDURE insertar_cuidador();
643
644
```

Figure 94: Creación trigger

3.13.2 Inserción y materialización del Trigger en la bitácora cuidador

```
732
733 select * from cuidadores;
734 INSERT into cuidadores (id_cuidador, fecha_ingreso_alparque) values(6, now());
735 select * from "BitacoraCuidadores";
736
```

	id_cuidador [PK] integer	fecha_ingreso_alparque date
1	133	2021-01-23
2	233	2019-07-23

Figure 95: Inserciones en tabla cuidadores

3.13.3 *SELECT BitacoraCuidadores*

732

733

734

735

736

Notifications

Data Output

Messages

Explain

id_cuidador_old

integer

fecha_ingreso_alparque_old

date

id_cuidador_new

integer

fecha_ingreso_alparque_new

date

usuario

character varying (15)

accion

character (1)

fechaRegistro

date

1

130

2021-01-23

133

2021-01-23

postgres

A

2021-06-25

2

132

2019-07-23

233

2019-07-23

postgres

A

2021-06-25

3

0

-infinity

6

2021-06-25

postgres

I

2021-06-25

4

6

2021-06-25

0

infinity

postgres

D

2021-06-25

5

0

-infinity

6

2021-06-25

postgres

I

2021-06-25

Figure 96: Seleccionando valores de la bitácora cuidadores

3.14 Tabla cuidador_especie

3.14.1 Trigger que llama a la función que registra en la Tabla Bitácora Cuidador Especie

```
141 -- insertar cuidador especie
142 CREATE FUNCTION insertar_cuidador_especie() RETURNS TRIGGER AS
143 $$
144 DECLARE
145
146     accion character(1) := 'I';
147     usuario varchar(30) := (select user);
148     fecha_registro date := now();
149     id_nulo integer := 0;
150     fecha_de_inicio_cuido_nulo date := 'infinity';
151     id_especies_nulo integer := 0;
152     id_cuidador_nulo integer := 0;
153
154
155 BEGIN
156
157     INSERT INTO public."BitacoraCuidador_especie"
158     VALUES (id_nulo, fecha_de_inicio_cuido_nulo,
159             id_especies_nulo, id_cuidador_nulo,
160
161             NEW.id_cuido, NEW.fecha_de_inicio_cuido,
162             NEW.id_especies, NEW.id_cuidador,
163
164             usuario, accion, fecha_registro);
165
166     RETURN NEW;
167
168 END
169 $$
170 LANGUAGE PLPGSQL;
```

Figure 97: Creación función trigger insertar cuidador especie

```
629
630 -- Trigger Insertar Cuidador especie
631 CREATE TRIGGER Trigger_InsertarCuidadorEspecie AFTER INSERT ON cuidador_especie
632 FOR EACH ROW EXECUTE PROCEDURE insertar_cuidador_especie();
633
```

Figure 98: Creación trigger

3.14.2 Inserción y materialización del Trigger en la bitácora cuidador especie

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

730

731

727

728

729

Figure 99: Inserciones en tabla cuidador_especie

3.14.3 *SELECT BitacoraCuidadorEspecie*

```

730 select * from cuidador_especie;
731 INSERT into cuidador_especie (id_cuido, fecha_de_inicio_cuido, id_especies, id_cuidador) values(6, now(), 50, 6);
732 select * from "BitacoraCuidador_especie";
733

```

Notifications	Data Output	Messages	Explain						
se_inicio_cuido_old	id_especies_old	id_cuidador_old	id_cuido_new	fecha_de_inicio_cuido_new	id_especies_new	id_cuidador_new	usuario	accion	fechaRegistrn
7-23	50	130	1	2015-07-23	50	133	postgres	A	2021-06-25
	0	0	6	2021-06-25	50	133	postgres	I	2021-06-25
5-25	50	133	0	infinity	0	0	postgres	D	2021-06-25
	0	0	6	2021-06-25	50	6	postgres	I	2021-06-25

Figure 100: Seleccionando valores de la bitácora cuidador especie

3.15 Tabla descripcion_especie

3.15.1 Trigger que llama a la función que registra en la Tabla Bitácora Descripción Especie

```
262 -- insertar descripcion especie
263 CREATE FUNCTION insertar_descripcion_especie() RETURNS TRIGGER AS
264 $$
265 DECLARE
266
267     accion character(1) := 'I';
268     usuario varchar(30) := (select user);
269     fecha_registro date := now();
270     id_nulo integer := 0;
271     sexo_nulo character(1) := ' ';
272     fecha_de_nacimiento_nulo date := '-infinity';
273     peso_nulo double precision := 0.0;
274
275
276 BEGIN
277
278     INSERT INTO public."BitacoraDescripcion_especie"
279     VALUES (id_nulo, sexo_nulo, fecha_de_nacimiento_nulo, peso_nulo,
280
281             NEW.id_descripcion, NEW.sexo, NEW.fecha_nacimiento, NEW.peso,
282
283             usuario, accion, fecha_registro);
284
285     RETURN NEW;
286
287 END
288 $$
289 LANGUAGE PLPGSQL;
```

Figure 101: Creación función trigger insertar descripcion especie

```
653
654 -- Trigger Insertar Descripción Especie
655 CREATE TRIGGER Trigger_InsertarDescripcionEspecie AFTER INSERT ON descripcion_especie
656 FOR EACH ROW EXECUTE PROCEDURE insertar_descripcion_especie();
657
658
```

Figure 102: Creación trigger

3.15.2 Inserción y materialización del Trigger en la bitácora descripción especie

757

758

759

760

761

select * from descripcion_especie;

INSERT into descripcion_especie (id_descripcion, sexo, fecha_nacimiento, peso) values(6, 'M', '2010-10-10', 30.20);

select * from "BitacoraDescripcion_especie";

Notifications

Data Output

Messages

Explain

	id_descripcion [PK] integer	sexo character (1)	fecha_nacimiento date	peso double precision
1	1000	M	2015-07-23	100.32

Figure 103: Inserciones en tabla descripcion especie

3.15.3 *SELECT BitacoraDescripcion_especie*

```
779 select * from descripcion_especie;
780 INSERT into descripcion_especie (id_descripcion, sexo, fecha_nacimiento, peso) values(6, 'M', '2010-10-10', 30.20);
781 select * from "BitacoraDescripcion_especie";
782
```

Notifications Data Output Messages Explain									
sexo_old character (1)	fecha_nacimiento_old date	peso_old double precision	id_descripcion_new integer	sexo_new character (1)	fecha_nacimiento_new date	peso_new double precision	usuario character varying (15)	accion character (1)	fechaRegistro date
M	2015-07-23	75.5	1000	M	2015-07-23	100.32	postgres	A	2021-06-23
	-infinity	0	6	M	2010-10-10	30.2	postgres	I	2021-06-25
M	2010-10-10	30.2	0		Infinity	0	postgres	D	2021-06-25

Figure 104: Seleccionando valores de la bitácora descripcion especie

3.16 Tabla especie

3.16.1 Trigger que llama a la función que registra en la Tabla Bitácora Especie

```
327 -- insertar especie
328 CREATE FUNCTION insertar_especie() RETURNS TRIGGER AS
329 $$
330 DECLARE
331
332     accion character(1) := 'I';
333     usuario varchar(30) := (select user);
334     fecha_registro date := now();
335     id_nulo integer := 0;
336     nombre_comun_nulo character(1) := ' ';
337     nombre_cientifico_nulo character(1) := ' ';
338     id_descripcion_nulo integer := 0;
339     id_zona_nulo integer := 0;
340
341
342
343 BEGIN
344
345     INSERT INTO public."BitacoraEspecie"
346     VALUES (id_nulo, nombre_comun_nulo, nombre_cientifico_nulo,
347             id_descripcion_nulo, id_zona_nulo,
348
349             NEW.id_especies, NEW.nombre_comun, NEW.nombre_cientifico,
350             NEW.id_descripcion, NEW.id_zona,
351
352             usuario, accion, fecha_registro);
353
354     RETURN NEW;
355
356 END
357 $$
358 LANGUAGE PLPGSQL;
```

Figure 105: Creación función trigger insertar especie

```
665 |
666 -- Trigger Insertar Especie
667 CREATE TRIGGER Trigger_InsertarEspecie AFTER INSERT ON especie
668 FOR EACH ROW EXECUTE PROCEDURE insertar_especie();
669
670
```

Figure 106: Creación trigger

3.16.2 Inserción y materialización del Trigger en la bitácora especie

```
771
772 select * from especie;
773
774 INSERT into especie (id_especies, nombre_comun, nombre_cientifico, id_descripcion, id_zona)
775     values(6, 'Rana Roja', 'Oophaga pumilio', 5, 5);
776
777 select * from "BitacoraEspecie";
778
```

Notifications	Data Output	Messages	Explain		
	id_especies [PK] integer	nombre_comun character varying (30)	nombre_cientifico character varying (30)	id_descripcion integer	id_zona integer
	1	50	gdfgdfgdf	sadfsdf	1000 5001

Figure 107: Inserciones en tabla especie

3.16.3 *SELECT BitacoraEspecie*

```

730 select * from cuidador_especie;
731 INSERT into cuidador_especie (id_cuido, fecha_de_inicio_cuido, id_especies, id_cuidador) values(6, now(), 50, 6);
732 select * from "BitacoraCuidador_especie";
733

```

Notifications Data Output Messages Explain									
se_inicio_cuido_old	id_especies_old	id_cuidador_old	id_cuido_new	fecha_de_inicio_cuido_new	id_especies_new	id_cuidador_new	usuario	accion	fechaRegistrn
	integer	integer	integer	date	integer	integer	character varying (15)	character (1)	date
7-23	50	130	1	2015-07-23	50	133	postgres	A	2021-06-25
	0	0	6	2021-06-25	50	133	postgres	I	2021-06-25
5-25	50	133	0	infinity	0	0	postgres	D	2021-06-25
	0	0	6	2021-06-25	50	6	postgres	I	2021-06-25

Figure 108: Seleccionando valores de la bitácora especie

3.17 Tabla especie_habitat

3.17.1 Trigger que llama a la función que registra en la Tabla Bitácora Especie Habitat

```
390 -- insertar especie_habitat
391 CREATE FUNCTION insertar_especie_habitat() RETURNS TRIGGER AS
392 $$
393     DECLARE
394
395         accion character(1) := 'I';
396         usuario varchar(30) := (select user);
397         fecha_registro date := now();
398         id_nulo integer := 0;
399         habitat_id_habitat_nulo integer := 0;
400         id_especie_habitat_nulo integer := 0;
401
402
403 BEGIN
404
405     INSERT INTO public."BitacoraEspecieHabitat"
406     VALUES (id_nulo, habitat_id_habitat_nulo, id_especie_habitat_nulo,
407
408             NEW.especie_id_habitat, NEW.habitat_id_habitat, NEW.id_especie_habitat,
409
410             usuario, accion, fecha_registro);
411
412     RETURN NEW;
413
414 END
415 $$
416 LANGUAGE PLPGSQL;
```

Figure 109: Creación función trigger insertar especie habitat

```
677
678 -- Trigger Insertar EspecieHabitat
679 CREATE TRIGGER Trigger_InsertarEspecieHabitat AFTER INSERT ON especie_habitat
680 FOR EACH ROW EXECUTE PROCEDURE insertar_especie_habitat();
681
682
```

Figure 110: Creación trigger

3.17.2 Inserción y materialización del Trigger en la bitácora especie habitat

```
788 select * from especie_habitat;
789 INSERT into especie_habitat (especie_id_habitat, habitat_id_habitat, id_especie_habitat) values(5, 5, 6);
790 select * from "BitacoraEspecie_Habitat";
791
```

Notifications

Data Output

Messages

Explain

	especie_id_habitat integer	habitat_id_habitat integer	id_especie_habitat [PK] integer	
1	50	337	5	
2	50	337	6	

Figure 111: Inserciones en tabla especie habitat

3.17.3 *SELECT BitacoraEspecie_Habitat*

```
788 select * from especie_habitat;
789 INSERT into especie_habitat (especie_id_habitat, habitat_id_habitat, id_especie_habitat) values(6, 337, 7);
790 select * from "BitacoraEspecie_Habitat";
791
```

Notifications	Data Output	Messages	Explain					
id_habitat_old	habitat_id_habitat_old integer	id_especie_habitat_old integer	especie_id_habitat_new integer	habitat_id_habitat_new integer	id_especie_habitat_new integer	usuario character varying (15)	accion character (1)	fechaRegistro date
	0	0	0	50	337	5 postgres	I	2021-06-25
	0	0	0	50	337	6 postgres	I	2021-06-25
	50	337	1	0	0	0 postgres	D	2021-06-25
	0	0	0	6	337	7 postgres	I	2021-06-25

Figure 112: Seleccionando valores de la bitácora especie habitat

3.18 Tabla habitat

3.18.1 Trigger que llama a la función que registra en la Tabla Bitácora Habitat

```
449 -- insertar habitat
450 CREATE FUNCTION insertar_habitat() RETURNS TRIGGER AS
451 $$
452 DECLARE
453
454     accion character(1) := 'I';
455     usuario varchar(30) := (select user);
456     fecha_registro date := now();
457     id_nulo integer := 0;
458     nombre_nulo character(1) := ' ';
459     id_tipo_habitat_nulo integer := 0;
460     id_continente_nulo integer := 0;
461
462 BEGIN
463
464     INSERT INTO public."BitacoraHabitat"
465     VALUES (id_nulo, nombre_nulo, id_tipo_habitat_nulo, id_continente_nulo,
466
467             NEW.id_habitat, NEW.nombre, NEW.id_tipo_habitat, NEW.id_continente,
468
469             usuario, accion, fecha_registro);
470
471     RETURN NEW;
472
473 END
474 $$
475 LANGUAGE PLPGSQL;
```

Figure 113: Creación función trigger insertar habitat

```
689
690 -- Trigger Insertar Habitat
691 CREATE TRIGGER Trigger_InsertarHabitat AFTER INSERT ON habitat
692 FOR EACH ROW EXECUTE PROCEDURE insertar_habitat();
693
694
```

Figure 114: Creación trigger

3.18.2 Inserción y materialización del Trigger en la bitácora habitat

801

802

803

804

805

select * from habitat;

INSERT into habitat (id_habitat, nombre, id_tipo_habitat, id_continente) values(6, 'Tropical', 5, 5);

select * from "BitacoraHabitat";

Notifications

Data Output

Messages

Explain

	id_habitat [PK] integer	nombre character varying (30)	id_tipo_habitat integer	id_continente integer
1	336	sdknfsdknfsdnf	10	255
2	337	sdfsdafd	10	255

Figure 115: Inserciones en tabla habitat

3.18.3 *SELECT BitacoraHabitat*

816	
817	<code>select * from habitat;</code>
818	<code>INSERT into habitat (id_habitat, nombre, id_tipo_habitat, id_continente) values(6, 'Tropical', 5, 5);</code>
819	<code>select * from "BitacoraHabitat";</code>
820	

Notifications	Data Output	Messages	Explain
old varying (30)	id_tipo_habitat_old integer	id_continente_old integer	id_habitat_new integer
	10	255	336
	0	0	6
	10	255	0
			nombre_new character varying (30)
			sdknfasknjednf
			id_tipo_habitat_new integer
			10
			id_continente_new integer
			255
			usuario character varying (15)
			postgres
			accion character (1)
			A
			fechaRegistro date
			I
			2021-06-25
			D
			2021-06-25

Figure 116: Seleccionando valores de la bitácora habitat

3.19 Tabla lista_dezonas_visitadas

3.19.1 Trigger que llama a la función que registra en la Tabla Bitácora Lista de zonas visitadas

```
509 -- insertar zona visitadas
510 CREATE FUNCTION insertar_lista_dezonas_visitadas() RETURNS TRIGGER AS
511 $$
512     DECLARE
513
514         accion character(1) := 'I';
515         usuario varchar(30) := (select user);
516         fecha_registro date := now();
517         id_nulo integer := 0;
518         codigo_nulo varchar(30) := ' ';
519         id_zonas_nulo integer := 0;
520
521
522     BEGIN
523
524         INSERT INTO public."BitacoraLista_dezonas_visitadas"
525             VALUES (id_nulo, codigo_nulo, id_zonas_nulo,
526
527                     NEW.id_tours, NEW.codigo, NEW.id_zona,
528
529                     usuario, accion, fecha_registro);
530
531         RETURN NEW;
532
533     END
534 $$
535 LANGUAGE PLPGSQL;
536
```

Figure 117: Creación función trigger insertar lista de zonas visitadas

```
701
702 -- Trigger Insertar Zona Visitadas
703 CREATE TRIGGER Trigger_InsertarZonaVisitada AFTER INSERT ON lista_dezonas_visitadas
704     FOR EACH ROW EXECUTE PROCEDURE insertar_lista_dezonas_visitadas();
705
```

Figure 118: Creación trigger

3.19.2 Inserción y materialización del Trigger en la bitácora lista de zonas visitadas

809

810

811

812

813

select * from lista_dezonas_visitadas;

INSERT INTO lista_dezonas_visitadas(id_tours, codigo, id_zona) values(6, 'b737', 5001);

select * from "BitacoraLista_dezonas_visitadas";

Notifications

Data Output

Messages

Explain

id_tours

[PK] integer

codigo

character varying (30)

id_zona

integer

1

700

b737

5001

Figure 119: Inserciones en tabla lista de zonas visitadas

3.19.3 *SELECT BitacoraLista_dezonas_visitadas*

```
810 select * from lista_dezonas_visitadas;
811 INSERT INTO lista_dezonas_visitadas(id_tours, codigo, id_zona) values(6, 'b737', 5001);
812 select * from "BitacoraLista_dezonas_visitadas";
813
```

	id_tours_old integer	codigo_old character varying (30)	id_zona_old integer	id_tours_new integer	codigo_new character varying (30)	id_zona_new integer	usuario character varying (15)	accion character (1)	fechaRegistro date
1	700	3000	5001	700	b737	5001	postgres	A	2021-06-23
2	0		0	6	b737	5001	postgres	I	2021-06-25
3	6	b737	5001	0		0	postgres	D	2021-06-25
4	0		0	6	b737	5001	postgres	I	2021-06-25

Figure 120: Seleccionando valores de la bitácora lista de zonas visitadas

3.20 Tabla tipo_habitat

3.20.1 Trigger que llama a la función que registra en la Tabla Bitácora Tipo Habitat

```
568 -- insertar tipo_habitat
569 CREATE FUNCTION insertar_tipo_habitat() RETURNS TRIGGER AS
570 $$
571     DECLARE
572
573         accion character(1) := 'I';
574         usuario varchar(30) := (select user);
575         fecha_registro date := now();
576         id_nulo integer := 0;
577         clima_nulo character := ' ';
578         tipo_vegetacion_nulo character(1) = ' ';
579
580
581     BEGIN
582
583         INSERT INTO public."BitacoraTipo_habitat"
584             VALUES (id_nulo, clima_nulo, tipo_vegetacion_nulo,
585
586                     NEW.id_tipo_habitat, NEW.clima, NEW.tipo_vegetacion,
587
588                     usuario, accion, fecha_registro);
589
590     RETURN NEW;
591
592     END
593 $$
594 LANGUAGE PLPGSQL;
595
```

Figure 121: Creación función trigger insertar tipo de habitat

```
713
714 -- Trigger Insertar Tipo Habitat
715 CREATE TRIGGER Trigger_InsertarTipoHabitat AFTER INSERT ON tipo_habitat
716     FOR EACH ROW EXECUTE PROCEDURE insertar_tipo_habitat();
717
718
```

Figure 122: Creación trigger

3.20.2 Inserción y materialización del Trigger en la bitácora tipo de habitat

```
825
826 select * from tipo_habitat;
827 INSERT into tipo_habitat (id_tipo_habitat, clima, tipo_vegetacion) values(6, 'Soleado', 'Humedo');
828 select * from "BitacoraTipo_habitat";
829
```

Notifications	Data Output	Messages	Explain						
	<table> <tr> <th>id_tipo_habitat [PK] integer</th> <th>clima character varying (30)</th> <th>tipo_vegetacion character varying (30)</th> </tr> <tr> <td>1</td> <td>10 tropical</td> <td>bcbvcbvcbv</td> </tr> </table>	id_tipo_habitat [PK] integer	clima character varying (30)	tipo_vegetacion character varying (30)	1	10 tropical	bcbvcbvcbv		
id_tipo_habitat [PK] integer	clima character varying (30)	tipo_vegetacion character varying (30)							
1	10 tropical	bcbvcbvcbv							

Figure 123: Inserciones en tabla tipo_habitat

3.20.3 *SELECT BitacoraTipo_habitat*

825

826

827

828

829

select * from tipo_habitat;

INSERT into tipo_habitat (id_tipo_habitat, clima, tipo_vegetacion) values(6, 'Soleado', 'Humedo');

select * from "BitacoraTipo_habitat";

Notifications

Data Output

Messages

Explain

id_tipo_habitat_old

integer

clima_old

character varying (30)

tipo_vegetacion_old

character varying (30)

id_tipo_habitat_new

integer

clima_new

character varying (30)

tipo_vegetacion_new

character varying (30)

usuario

character varying (15)

accion

character (1)

fechaRegistro

date

1

10

tropical

sabana

10

tropical

bcbcbcbcbcb

postgres

A

2021-06-23

2

0

6

Soleado

Humedo

0

Soleado

Humedo

postgres

I

2021-06-25

3

6

Soleado

Humedo

0

postgres

D

2021-06-25

4

0

6

Soleado

Humedo

6

Soleado

Humedo

postgres

I

2021-06-25

Figure 124: Seleccionando valores de la bitácora tipo habitat

3.21 Tabla zona

3.21.1 Trigger que llama a la función que registra en la Tabla Bitácora Zona

```
81 -- insertar zona
82 CREATE FUNCTION insertar_zona() RETURNS TRIGGER AS
83 $$
84 DECLARE
85
86     accion character(1) := 'I';
87     usuario varchar(30) := (select user);
88     fecha_registro date := now();
89     id_nulo integer := 0;
90     nombre_nulo character(1) := ' ';
91     extension_territorial_nulo double precision := 0.0;
92
93 BEGIN
94
95     INSERT INTO public."BitacoraZona"
96     VALUES (id_nulo, nombre_nulo, extension_territorial_nulo,
97
98             NEW.id_zona, NEW.nombre, NEW.extension_territorial,
99
100             usuario, accion, fecha_registro);
101
102     RETURN NEW;
103
104 END
105 $$
106 LANGUAGE PLPGSQL;
```

Figure 125: Creación función trigger insertar zona

```
619
620 -- Trigger Insertar Zona
621 CREATE TRIGGER Trigger_InsertarZona AFTER INSERT ON zona
622     FOR EACH ROW EXECUTE PROCEDURE insertar_zona();
623
624
```

Figure 126: Creación trigger

3.21.2 Inserción y materialización del Trigger en la bitácora tipo de zona

```
738
739 select * from zona;
740 INSERT into zona (id_zona, nombre, extension_territorial) values(6, 'El Ecuador', 50.6);
741 select * from "BitacoraZona";
742
```

Notifications	Data Output	Messages	Explain
	id_zona [PK] integer	nombre character varying (30)	extension_territorial double precision
	1	5001 Central	1555.2

Figure 127: Inserciones en tabla zona

3.21.3 *SELECT BitacoraZona*

```
739 select * from zona;
740 INSERT into zona (id_zona, nombre, extension_territorial) values(6, 'El Ecuador', 50.6);
741 select * from "BitacoraZona";
742
```

Notifications	Data Output	Messages	Explain						
id_zona_old integer	nombre_old character varying (30)	extension_territorial_old double precision	id_zona_new integer	nombre_new character varying (30)	extension_territorial_new double precision	usuario character varying (15)	accion character (1)	fechaRegistro date	
1	5001	VIP	1555.2	5001	Central	1555.2	postgres	A	2021-06-24
2	0		0	6	Sur	506	postgres	I	2021-06-25
3	6	Sur	506	0		0	postgres	D	2021-06-25
4	0		0	6	El Ecuador	50.6	postgres	I	2021-06-25

Figure 128: Seleccionando valores de la bitácora zona