

## Contenido

1	Prueba de la materialización de los TRIGGERS con la sentencia Eliminar .....	3
1.1	Trigger que llama a la función que registra en la Tabla Bitácora Pais.....	3
1.2	Eliminación y materialización del Trigger en la bitácora pais .....	5
2	Trigger que llama a la función que registra en la Tabla Bitácora Provincia .....	6
2.1	Eliminación y materialización del Trigger en la bitácora provincia.....	7
3	Trigger que llama a la función que registra en la Tabla Bitácora Cantón .....	8
3.1	Eliminación y materialización del Trigger en la bitácora Canton .....	9
4	Trigger que llama a la función que registra en la Tabla Bitácora Distrito .....	10
4.1	Eliminación y materialización del Trigger en la bitácora Distrito.....	12
5	Trigger que llama a la función que registra en la Tabla Bitácora Barrio .....	13
5.1	Eliminación y materialización del Trigger en la bitácora Barrio.....	14
6	Trigger que llama a la función que registra en la Tabla Bitácora Empleado .....	16
6.1	Eliminación y materialización del Trigger en la bitácora Empleado .....	18
7	Trigger que llama a la función que registra en la Tabla Bitácora Contacto .....	19
7.1	Eliminación y materialización del Trigger en la bitácora Contacto .....	20
8	Trigger que llama a la función que registra en la Tabla Bitácora Guia.....	22
8.1	Eliminación y materialización del Trigger en la bitácora Contacto .....	23
9	Trigger que llama a la función que registra en la Tabla Bitácora Duración recorrido .....	25
9.1	Eliminación y materialización del Trigger en la bitácora Duración recorrido .....	26
10	Trigger que llama a la función que registra en la Tabla Bitácora Itinerarios .....	28
10.1	Eliminación y materialización del Trigger en la bitácora Itinerarios .....	29
11	Trigger que llama a la función que registra en la Tabla Bitácora Asignaciones .....	31
11.1	Eliminación y materialización del Trigger en la bitácora Asignaciones.....	32
12	eliminar_continente.....	34
12.1	Desplegando tabla continente .....	35
12.2	Seleccionando la Bitácora Continente .....	35
13	eliminar_cuidador .....	36
13.1	Desplegando tabla cuidadores.....	37
13.2	Seleccionando la bitácora cuidadores.....	37
14	eliminar_cuidador_especie .....	38
14.1	Desplegando tabla cuidador_especie .....	39

14.2	Seleccionando la bitácora cuidador especie, se puede ver como en la última acción (eliminar) fue guardado en la bitácora. Y esta acción es representada con el carácter <i>D</i> .....	40
15	eliminar_descripcion_especie .....	41
15.1	Desplegando la tabla descripcion_especie .....	42
15.2	Seleccionando la bitácora descripción especie .....	42
16	eliminar_especie .....	43
16.1	Desplegando la tabla especie .....	44
16.2	Desplegando bitácora especie, en la última acción se puede observar el carácter <i>D</i> para <i>Delete</i> , luego de haber eliminado una entrada de la tabla especie.....	45
17	eliminar_especie_habitat.....	46
17.1	Desplegando la tabla .....	47
17.2	Desplegando bitácora especie_habitat.....	47
18	<b>eliminar_habitat</b> .....	48
18.1	Desplegando la tabla habitat .....	49
	.....	49
18.2	Desplegando la bitacora habitat .....	49
19	eliminar_lista_dezonas_visitadas.....	50
19.1	Desplegando la tabla lista de zonas visitadas .....	51
19.2	Desplegando la bitácora lista de zonas visitadas .....	52
20	<b>eliminar_tipo_habitat</b> .....	53
20.1	Desplegando la tabla tipo hábitat .....	54
20.2	Desplegando la bitácora tipo habitat.....	54
21	eliminar_zona.....	55
21.1	Desplegando la tabla zona .....	56
21.2	Desplegando la bitácora zona .....	57

## 1 Prueba de la materialización de los TRIGGERS con la sentencia Eliminar

### 1.1 Trigger que llama a la función que registra en la Tabla Bitácora Pais

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

```
INSERT INTO public."BitacoraPais"  
-- 'D' Hace referencia a la sentencia DELETE  
VALUES (vid_pais,vnombre,0,' ',vusuario,'D',now());  
  
RETURN NEW;  
  
END  
  
$$;  
  
CREATE TRIGGER Trigger_TablaPaisELIMINAR  
AFTER DELETE  
ON public.pais  
--CALL EL TRIGGER  
FOR EACH ROW EXECUTE PROCEDURE ControlEliminarPais();
```

## 1.2 Eliminacion y materialización del Trigger en la bitácora pais

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
52	Mexico	0		postgres	D	2021-06-24
55	Brasil	0		postgres	D	2021-06-24

## 2 Trigger que llama a la función que registra en la Tabla Bitácora Provincia

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

```
DECLARE
```

```
vusuario varchar(15);  
vid_provincia integer;  
vnombre character(30);  
vid_pais integer;  
BEGIN  
vid_provincia =OLD.id_provincia;  
vnombre=OLD.nombre;  
vid_pais=OLD.id_pais;  
SELECT user INTO vusuario;
```

```
INSERT INTO public."BitacoraProvincia"
```

```
-- 'D' Hace referencia a la sentencia DELETE
```

```
VALUES (vid_provincia,vnombre,vid_pais,0 ,'',0,vusuario,'D',now());
```

```
RETURN NEW;
```

```
END
```

```
$$;
```

```
CREATE TRIGGER Trigger_TablaProvinciaELIMINAR  
AFTER DELETE
```

```

ON public.provincia|
--CALL EL TRIGGER

FOR EACH ROW EXECUTE PROCEDURE    ControlEliminarProvincia();

```

## 2.1 Eliminacion y materialización del Trigger en la bitácora provincia

```

DELETE FROM public.provincia
WHERE id_provincia = '8';

SELECT id_provincia, nombre, id_pais
FROM public.provincia;

```

id_provincia_old	nombre_old	id_pais_old	id_provincia_new	nombre_new	id_pais_new	usuario	accion	fechaRegistro
integer	character varying (30)	integer	integer	character varying (30)	integer	character varying (15)	character (1)	date
1	San Jose	506	1	Chepe	506	postgres	A	2021-06-23
8	Limon	506	0		0	postgres	D	2021-06-24

### 3 Trigger que llama a la función que registra en la Tabla Bitácora Cantón

```
CREATE FUNCTION ControlEliminarCanton()  
returns TRIGGER  
LANGUAGE PLPGSQL  
AS $$  
  
DECLARE  
vusuario varchar(15);  
vid_canton integer;  
vnombre character(30);  
vid_provincia integer;  
vid_pais integer;  
BEGIN  
vid_canton =OLD.id_canton;  
vnombre=OLD.nombre;  
vid_provincia =OLD.id_provincia;  
vid_pais=OLD.id_pais;  
SELECT user INTO vusuario;  
  
INSERT INTO public."BitacoraCanton"  
  
VALUES (vid_canton,vnombre,vid_provincia,vid_pais,0 ,',',0,0,vusuario,'D',now());  
  
RETURN NEW;  
  
END  
  
$$;  
  
CREATE TRIGGER Trigger_TablaCantonELIMINAR  
AFTER DELETE  
  
ON public.canton  
--CALL EL TRIGGER  
  
FOR EACH ROW EXECUTE PROCEDURE ControlEliminarCanton();
```



### 3.1 Eliminación y materialización del Trigger en la bitácora Canton

```
DELETE FROM public.canton
WHERE id_canton= '3';
```

```
DELETE FROM public.canton
WHERE id_canton = '4';
```

id_canton_old integer	nombre_old character varying (30)	id_provincia_old integer	id_canton_new integer	nombre_new character varying (30)	id_provincia_new integer	usuario character varying (15)	accion character (1)	fechaRegist date
1	San Jose	1	1	Sabana	1	postgres	A	2021-06-23
3	Paxton	3	0		0	postgres	D	2021-06-24
4	Liva	4	0		0	postgres	D	2021-06-24

#### 4 Trigger que llama a la función que registra en la Tabla Bitácora Distrito

```
CREATE FUNCTION ControlEliminarDistrito()  
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 =OLD.id_canton;  
vnombre=OLD.nombre;  
vid_provincia =OLD.id_provincia;  
vid_distrito=OLD.id_distrito;
```

```
vid_pais =OLD.id_pais;
SELECT user INTO vusuario;

INSERT INTO public."BitacoraDistrito"

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

RETURN NEW;

END

$$;

CREATE TRIGGER Trigger_TablaDistritoELIMINAR
AFTER DELETE

ON public.distrito
```

#### 4.1 Eliminación y materialización del Trigger en la bitácora Distrito

```
DELETE FROM public.distrito
WHERE id_distrito = '5';
```

	id_distrito_old integer	nombre_old character varying (30)	id_canton_old integer	id_provincia_old integer	id_distrito_new integer	nombre_new character varying (30)	id_canton_new integer	id_provincia_new integer	usuario character varying (15)	accion character (1)	fechaRegistro date
1	5	Ambrosius	5	5	0		0	0	postgres	I	2021-06-24

## 5 Trigger que llama a la función que registra en la Tabla Bitácora Barrio

```
CREATE FUNCTION ControlEliminarBarrio()  
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 =OLD.id_canton;  
vnombre=OLD.nombre;  
vid_provincia =OLD.id_provincia;  
vid_distrito=OLD.id_distrito;
```

```

vid_distrito=OLD.id_distrito;
vid_barrio=OLD.id_barrio;
vid_pais=OLD.id_pais;
SELECT user INTO vusuario;

INSERT INTO public."BitacoraBarrio"

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

RETURN NEW;

END

$$;

CREATE TRIGGER Trigger_TablaBarrioELIMINAR
AFTER DELETE

ON public.barrio

```

## 5.1 Eliminación y materialización del Trigger en la bitácora Barrio

```

DELETE FROM public.barrio
WHERE id_barrio = '2';

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

```

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

	usuario character varying (15)	accion character (1)	fechaRegistro date
7	postgres	A	2021-06-23
0	postgres	D	2021-06-24

## 6 Trigger que llama a la función que registra en la Tabla Bitácora Empleado

```
CREATE FUNCTION ControlEliminarEmpleado()  
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;  
vid_pais integer;
```

```
9 BEGIN  
1 vid_empleado=OLD.id_empleado;  
2 vprimer_nombre=OLD.primer_nombre;  
3 vsegundo_nombre =OLD.segundo_nombre;  
4 vprimer_apellido =OLD.primer_apellido;  
5 vsegundo_apellido=OLD.segundo_apellido;  
6 vid_canton =OLD.id_canton;  
7  
8 vid_provincia =OLD.id_provincia;  
9 vid_distrito=OLD.id_distrito;  
9 vid_barrio=OLD.id_barrio;  
1 vid_pais=OLD.id_pais;  
2 SELECT user INTO vusuario;  
3  
4 INSERT INTO public."BitacoraEmpleados"
```



```
VALUES (vid_empleado,vprimer_nombre,vsegundo_nombre,vprimer_apellido,vsegundo_apellido,vid_provincia,vid_canton ,vid_distrito,\
```

```
,vid_barrio,vid_pais,0,' ',' ',' ',' ',0,0,0,0,0,vusuario,'D',now());
```

```
RETURN NEW;
```

```
END
```

```
$$;
```

```
CREATE TRIGGER Trigger_TablaEmpleadoEliminar  
AFTER DELETE
```

```
ON public.empleados
```

```
--CALL EL TRIGGER
```

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

## 6.1 Eliminación y materialización del Trigger en la bitácora Empleado

```
DELETE FROM public.empleados  
WHERE id_empleado = '1234'; |
```

2077	dorian	[null]	Rodriguez	fdfsdf	7	56	100	500
2077	dorian	[null]	sjdfbsjfb sdf	fdfsdf	7	56	100	500
2077	dorian	[null]	Rodriguez	fdfsdf	7	56	100	500
2077	dorian	[null]	Rodriguez	fdfsdf	7	56	100	500
1234	Julian	Daniel	ordonez	Guzman	1	1	1	1

postgres

D

2021-06-24

## 7 Trigger que llama a la función que registra en la Tabla Bitácora Contacto

```
CREATE FUNCTION ControlEliminarContacto()
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 =OLD.id_contacto;
vid_empleado =OLD.id_empleado;

vnumero_de_telefono=OLD.numero_de_telefono;
vnumero_secundario =OLD.numero_secundario;
vcorreo_electronico=OLD.correo_electronico;
vcorreo_electronico_secundario=OLD.correo_electronico_secundario;
SELECT user INTO vusuario;

INSERT INTO public."BitacoraContacto"

VALUES (vid_contacto ,vid_empleado ,vnumero_de_telefono,vnumero_secundario,vcorreo_electronico,vcorreo_electronico_secundario,0,0,'','','',

,vusuario, 'D' ,now()) ;
```

```
RETURN NEW;  
  
END  
  
$$;  
  
CREATE TRIGGER Trigger_TablaContactoEliminar  
AFTER DELETE  
  
ON public.contacto  
--CALL EL TRIGGER  
  
FOR EACH ROW EXECUTE PROCEDURE ControlEliminarContacto();  
  
--trigger de insercion de la tabla guia
```

### 7.1 Eliminación y materialización del Trigger en la bitácora Contacto

```
DELETE FROM public.contacto  
WHERE id_contacto= '15';
```

id_contacto_old integer	id_empleado_old integer	numero_de_telefono_old character (30)	numero_secundario_old character (30)	correo_electronico_old character (60)	correo_electronico_secundario_old character (60)	id_contacto_new integer	id_empleado_new integer
3	7882	245544	85222	sdfdd@mail.com	sdfsd@mail2.com	3	7882
0	0					4	5001
0	0					5	2004
0	0					10	5003
0	0					15	2001
15	2001	131 387 4436	492-215-0061	jdowzell2@newyorker.com	cpehrsson2@amazon.co.uk	0	

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 (15)	accion character (1)	fechaRegistro date
7882	1554564565465	85222	sdfsd@mail2.com	sdfsd@mail2.com	postgres	A	2021-06-23
5001	401 964 5175	564-568-9916	knibley3@bloomberg.com	hziemen3@ebay.co.uk	postgres	I	2021-06-23
2004	250 223 5071	149-554-0359	prubury4@china.com.cn	syushankin4@hc360.com	postgres	I	2021-06-23
5003	221 346 4883	494-934-3852	tduckham1@ustream.tv	bshuttlewood1@mayoclinic.com	postgres	I	2021-06-23
2001	131 387 4436	492-215-0061	jdowzell2@newyorker.com	cpehrsson2@amazon.co.uk	postgres	I	2021-06-23
0					postgres	D	2021-06-24

## 8 Trigger que llama a la función que registra en la Tabla Bitácora Guia

```
CREATE FUNCTION ControlEliminarGuia()  
returns TRIGGER  
LANGUAGE PLPGSQL  
AS $$  
  
DECLARE  
  
vusuario varchar(15);  
vid_guia integer;  
vid_empleado integer;  
vfecha_deingreso DATE;  
  
BEGIN  
vid_guia =OLD.id_guia;  
vid_empleado =OLD.id_empleado;  
vfecha_deingreso=OLD.fecha_deingreso_laboral;  
  
SELECT user INTO vusuario;
```

```

VALUES (vid_guia,vid_empleado,vfecha_deingreso,0,0,null,vusuario,'D',now());

RETURN NEW;

END

$$;

CREATE TRIGGER Trigger_TablaGuiaEliminar
AFTER DELETE

ON public.guia
--CALL EL TRIGGER

FOR EACH ROW EXECUTE PROCEDURE ControlEliminarGuia();

```

### 8.1 Eliminación y materialización del Trigger en la bitácora Contacto

```

DELETE FROM public.guia
WHERE id_guia= '4' ;

```





## 9 Trigger que llama a la función que registra en la Tabla Bitácora Duración recorrido

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

DECLARE

vusuario varchar(15);
vid_duracion_recorrido integer;
vhora integer;
vminutos integer;
vsegundos integer;

BEGIN
vid_duracion_recorrido =OLD.id_duracion_recorrido;
vhora =OLD.hora;

vminutos=OLD.minutos;
vsegundos=OLD.segundos;

SELECT user INTO vusuario;

INSERT INTO public."BitacoraDuracion_recorrido"

VALUES (vid_duracion_recorrido,vhora,vminutos,vsegundos,0,0,0,0,vusuario,'D',now());

RETURN NEW;

END

$$;


CREATE TRIGGER Trigger_TablaDuracion_Recorrido_Eliminar
AFTER DELETE
```

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

### 9.1 Eliminación y materialización del Trigger en la bitácora Duración recorrido

```
DELETE FROM public.duraccion_recorrido
WHERE id_duracion_recorrido= '3' ;
```

id_duracion_recorrido_old integer	hora_old integer	minutos_old integer	segundos_old integer	id_duracion_recorrido_new integer	hora_new integer	minutos_new integer	segundos_new integer	usuario character varying (15)	accion character (1)
90	1	0	0	90	1	30	0	postgres	A
0	0	0	0	1	11	1	0	postgres	I
0	0	0	0	2	5	14	0	postgres	I
0	0	0	0	3	5	47	0	postgres	I
3	5	47	0	0	0	0	0	postgres	D

fechaRegistro date	
2021-06-23	
2021-06-23	
2021-06-23	
2021-06-23	
2021-06-24	

## 10 Trigger que llama a la función que registra en la Tabla Bitácora Itinerarios

```
CREATE FUNCTION ControlEliminarItinerarios()  
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;
```

```
BEGIN
```

```
BEGIN
```

```
vcodigo=OLD.codigo;
```

```
vid_duracion_recorrido =OLD.id_duracion_recorrido;
```

```
vlongitud_recorrido =OLD.longitud_recorrido;
```

```
vnumero_maximodevisitante_autorizados=OLD.numero_maximodevisitante_autorizados;
```

```
vnumero_dedistintasespecies_quevisita =OLD.numero_dedistintasespecies_quevisita ;
```

```
SELECT user INTO vusuario;
```

```
INSERT INTO public."BitacoraItinerarios"
```

```
VALUES (vcodigo,vid_duracion_recorrido,vlongitud_recorrido,vnumero_maximodevisitante_autorizados,vnumero_dedistintasespecies_quevisita,  
|,0,0,0,0,vusuario,'I',now());
```

```
RETURN NEW;
```

```

RETURN NEW;

END

$$;

CREATE TRIGGER Trigger_TablaItinerarios_Eliminar
AFTER DELETE

ON public.itinerarios
--CALL EL TRIGGER

FOR EACH ROW EXECUTE PROCEDURE ControlEliminarItinerarios();

```

## 10.1 Eliminación y materialización del Trigger en la bitácora Itinerarios

```

DELETE FROM public.itinerarios
WHERE codigo= 'GH30' ;

```

<b>codigo_old</b> character varying (30)	<b>id_duracion_recorrido_old</b> integer	<b>longitud_recorrido_old</b> double precision	<b>numero_maximodevisitante_autorizados_old</b> integer	<b>numero_dedistintasespecies_quevisita_old</b> integer
3000	90	100.31	10	12
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
GH30	1	300.4	33	32
GH30	1	300.4	33	32

<b>usuario</b> character varying (15)	<b>accion</b> character (1)	<b>fechaRegistro</b> date
postgres	A	2021-06-23
postgres	I	2021-06-23
postgres	I	2021-06-23
postgres	I	2021-06-23
postgres	I	2021-06-23
postgres	I	2021-06-24
postgres	D	2021-06-24

## 11 Trigger que llama a la función que registra en la Tabla Bitácora Asignaciones

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

```
DECLARE
```

```
vusuario varchar(15);  
vid_asignaciones integer;  
vid_guia integer;
```

```
vhora_comienzo TIME;  
vcodigo varchar(30);
```

```
BEGIN
```

```
vid_guia =OLD.id_guia;  
vhora_comienzo =OLD.hora_comienzo;  
vcodigo =OLD.codigo ;
```

```
SELECT user INTO vusuario;
```

```
INSERT INTO public."BitacoraAsignaciones"
```

```
VALUES (vid_asignaciones,vhora_comienzo ,vid_guia,vcodigo,0,null,0,',',vusuario,'D',now());
```

```
RETURN NEW;
```

```
END
```

```
$$;
```

```
CREATE TRIGGER Trigger_TablaEliminarAsignaciones
AFTER DELETE

ON public.asignaciones
--CALL EL TRIGGER

FOR EACH ROW EXECUTE PROCEDURE ControlEliminarAsignaciones();

--fin del trigger de control de eliminacion de asignaciones
```

### 11.1 Eliminación y materialización del Trigger en la bitácora Asignaciones

```
DELETE FROM public.asignaciones
WHERE id_asignaciones= '10' ;
```



id_asignaciones_old integer	hora_comienzo_old time without time zone	id_guia_old integer	codigo_old character varying (30)	id_asignaciones_new integer	hora_comienzo_new time without time zone	id_guia_new integer	codigo_new character varying (30)	usuario character varying
1	07:00:00	900	3000	1	07:30:00	900	3000	postgres
0	[null]	0		4	11:23:02	1	GH22	postgres
0	[null]	0		5	10:52:01	2	BH26	postgres
0	[null]	0		10	10:50:01	2	BH26	postgres
10	10:50:01	2	BH26	0	[null]	0		postgres

accion character (1)	fechaRegistro date
A	2021-06-23
I	2021-06-23
I	2021-06-23
I	2021-06-23
D	2021-06-24

## 12 eliminar\_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;
```

## 12.1 Desplegando tabla continente

```
709
710 SELECT * FROM continente;
711 DELETE FROM continente WHERE id_continente = 2;
712 SELECT * FROM "BitacoraContinente";
713
714
715
716
717
718
```

	Notifications	Data Output	Messages	Explain
		id_continente [PK] integer	nombre character varying (30)	
1		255	Europa	
2		2	Antartica	

## 12.2 Seleccionando la Bitácora Continente

```
709
710 SELECT * FROM continente;
711 DELETE FROM continente WHERE id_continente = 2;
712 SELECT * FROM "BitacoraContinente";
713
714
```

Notifications

Data Output

Messages

Explain

	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

## 13 eliminar\_cuidador

```
172
173 -- eliminar cuidador
174 CREATE FUNCTION eliminar_cuidador() RETURNS TRIGGER AS
175 $$
176     DECLARE
177
178         accion character(1) := 'D';
179         usuario varchar(30) := (select user);
180         fecha_registro date := now();
181         id_nulo integer := 0;
182         fecha_de_ingreso_alparque_nulo date := 'infinity';
183
184
185
186     BEGIN
187
188         INSERT INTO public."BitacoraCuidadores"
189             VALUES (OLD.id_cuidador, OLD.fecha_ingreso_alparque,
190
191                     id_nulo, fecha_de_ingreso_alparque_nulo,
192
193                     usuario, accion, fecha_registro);
194
195         RETURN NEW;
196
197     END
198 $$
199 LANGUAGE PLPGSQL;
```

### 13.1 Desplegando tabla cuidadores

```
747 select * from cuidadores;
748
749 DELETE FROM cuidadores where id_cuidador = 6;
750
751 select * from "BitacoraCuidadores";
752
```

	id_cuidador [PK] integer	fecha_ingreso_alparque date
1	130	2021-01-23
2	132	2019-07-23
3	6	2021-06-25

### 13.2 Seleccionando la bitácora cuidadores

```
747 select * from cuidadores;
748
749 DELETE FROM cuidadores where id_cuidador = 6;
750
751 select * from "BitacoraCuidadores";
752
```

	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

## 14 eliminar\_cuidador\_especie

```
109 -- eliminar cuidador especie
110 CREATE FUNCTION eliminar_cuidador_especie() RETURNS TRIGGER AS
111 $$
112     DECLARE
113
114         accion character(1) := 'D';
115         usuario varchar(30) := (select user);
116         fecha_registro date := now();
117         id_nulo integer := 0;
118         fecha_de_inicio_cuido_nulo date := 'infinity';
119         id_especies_nulo integer := 0;
120         id_cuidador_nulo integer := 0;
121
122
123     BEGIN
124
125         INSERT INTO public."BitacoraCuidador_especie"
126             VALUES (OLD.id_cuido, OLD.fecha_de_inicio_cuido,
127                     OLD.id_especies, OLD.id_cuidador ,
128
129                     id_nulo, fecha_de_inicio_cuido_nulo,
130                     id_especies_nulo, id_cuidador_nulo,
131
132                     usuario, accion, fecha_registro);
133
134     RETURN NEW;
135
136     END
137 $$
138 LANGUAGE PLPGSQL;
```

## 14.1 Desplegando tabla cuidador\_especie

```
747  
748 select * from cuidador_especie;  
749 DELETE FROM cuidador_especie where id_cuido= 6;  
750 select * from "BitacoraCuidador_especie";  
751
```

	Notifications	Data Output	Messages	Explain
	id_cuido [PK] integer	fecha_de_inicio_cuido date	id_especies integer	id_cuidador integer
1	1	2015-07-23	50	133
2	6	2021-06-25	50	133

14.2 Seleccionando la bitácora cuidador especie, se puede ver como en la última acción (eliminar) fue guardado en la bitácora. Y esta acción es representada con el carácter *D*

```
748 select * from cuidador_especie;  
749 DELETE FROM cuidador_especie where id_cuido= 6;  
750 select * from "BitacoraCuidador_especie";  
751
```

Notifications Data Output Messages Explain

_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	fechaRegistro
	integer	integer	integer	date	integer	integer	character varying (15)	character (1)	date
3	50	132	1	2015-07-23	50	130	postgres	A	2021-06-23
3	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	50	133	0	infinity	0	0	postgres	D	2021-06-25



## 15 eliminar\_descripcion\_especie

```
230
231 -- eliminar descripcion especie
232 CREATE FUNCTION eliminar_descripcion_especie() RETURNS TRIGGER AS
233 $$
234     DECLARE
235
236         accion character(1) := 'D';
237         usuario varchar(30) := (select user);
238         fecha_registro date := now();
239         id_nulo integer := 0;
240         sexo_nulo character(1) := ' ';
241         fecha_de_nacimiento_nulo date := 'infinity';
242         peso_nulo double precision := 0.0;
243
244
245
246     BEGIN
247
248         INSERT INTO public."BitacoraDescripcion_especie"
249             VALUES (OLD.id_descripcion, OLD.sexo, OLD.fecha_nacimiento, OLD.peso,
250
251                     id_nulo, sexo_nulo, fecha_de_nacimiento_nulo, peso_nulo,
252
253                     usuario, accion, fecha_registro);
254
255     RETURN NEW;
256
257     END
258 $$
259 LANGUAGE PLPGSQL;
```

## 15.1 Desplegando la tabla descripcion\_especie

```
800 select * from descripcion_especie;
801 DELETE FROM descripcion_especie where id_descripcion = 6;
802 select * from "BitacoraDescripcion_especie";
803
```

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
2	6	M	2010-10-10	30.2

## 15.2 Seleccionando la bitácora descripción especie

```
800 select * from descripcion_especie;
801 DELETE FROM descripcion_especie where id_descripcion = 6;
802 select * from "BitacoraDescripcion_especie";
803
```

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

## 16 eliminar\_especie

```
294 -- eliminar especie
295 CREATE FUNCTION eliminar_especie() RETURNS TRIGGER AS
296 $$
297     DECLARE
298
299         accion character(1) := 'D';
300         usuario varchar(30) := (select user);
301         fecha_registro date := now();
302         id_nulo integer := 0;
303         nombre_comun_nulo character(1) := ' ';
304         nombre_cientifico_nulo character(1) := ' ';
305         id_descripcion_nulo integer := 0;
306         id_zona_nulo integer := 0;
307
308
309
310     BEGIN
311
312         INSERT INTO public."BitacoraEspecie"
313             VALUES (OLD.id_especies, OLD.nombre_comun, OLD.nombre_cientifico,
314                     OLD.id_descripcion, OLD.id_zona,
315
316                     id_nulo, nombre_comun_nulo, nombre_cientifico_nulo,
317                     id_descripcion_nulo, id_zona_nulo,
318
319                     usuario, accion, fecha_registro);
320
321     RETURN NEW;
322
323     END
324 $$
325 LANGUAGE PLPGSQL;
```

## 16.1 Desplegando la tabla especie

818







```
819 select * from especie;
```

```
820 -- DELETE FROM especie where id_especies = 6;
```

```
821 -- select * from "BitacoraEspecie";
```

822

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	sadfsdfsdf	1000	5001
2	6	Rana Roja	Oophaga pumilio	1000	5001

16.2 Desplegando bitácora especie, en la última acción se puede observar el carácter D para *Delete*, luego de haber eliminado una entrada de la tabla especie

```
818
819 select * from especie;
820 DELETE FROM especie where id_especies = 6;
821 select * from "BitacoraEspecie";
822
```

Notifications Data Output Messages Explain

id_descripcion_old integer	id_zona_old integer	id_especies_new integer	nombre_comun_new character varying (30)	nombre_cientifico_new character varying (30)	id_descripcion_new integer	id_zona_new integer	usuario character varying (15)	accion character (1)	fechaRegistro date
1000	5001	50	mariposa	sadfsdfsdf	1000	5001	postgres	A	2021-06-23
1000	5001	50	gdfgdfgdf	sadfsdfsdf	1000	5001	postgres	A	2021-06-23
0	0	6	Rana Roja	Oophaga pumilio	1000	5001	postgres	I	2021-06-25
1000	5001	0			0	0	postgres	D	2021-06-25

## 17 eliminar\_especie\_habitat

```
360
361 -- eliminar especie_habitat
362 CREATE FUNCTION eliminar_especie_habitat() RETURNS TRIGGER AS
363 $$
364     DECLARE
365
366         accion character(1) := 'D';
367         usuario varchar(30) := (select user);
368         fecha_registro date := now();
369         id_nulo integer := 0;
370         habitat_id_habitat_nulo integer := 0;
371         id_especie_habitat_nulo integer := 0;
372
373
374     BEGIN
375
376         INSERT INTO public."BitacoraEspecie_Habitat"
377             VALUES (OLD.especie_id_habitat, OLD.habitat_id_habitat, OLD.id_especie_habitat,
378
379
380                 id_nulo, habitat_id_habitat_nulo, id_especie_habitat_nulo,
381
382                 usuario, accion, fecha_registro);
383
384         RETURN NEW;
385
386     END
387 $$
388 LANGUAGE PLPGSQL;
```

## 17.1 Desplegando la tabla

```

842
843 select * from especie_habitat;
844 DELETE FROM especie_habitat where id_especie_habitat = 1;
845 select * from "BitacoraEspecie_Habitat";
846

```

Notifications Data Output Messages Explain

	especie_id_habitat integer	habitat_id_habitat integer	id_especie_habitat [PK] integer	
1	50	337	1	
2	50	337	5	
3	50	337	6	

## 17.2 Desplegando bitácora especie\_habitat

```

843 select * from especie_habitat;
844 DELETE FROM especie_habitat where id_especie_habitat = 1;
845 select * from "BitacoraEspecie_Habitat";
846

```

Notifications Data Output Messages Explain

	especie_id_habitat_old integer	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)	fecha date
1	1	336	1	1	337	1	postgres	A	202
2	0	0	0	50	337	5	postgres	I	202
3	0	0	0	50	337	6	postgres	I	202
4	50	337	1	0	0	0	postgres	D	202

## 18eliminar\_habitat

```
419 -- eliminar habitat
420 CREATE FUNCTION eliminar_habitat() RETURNS TRIGGER AS
421 $$
422     DECLARE
423
424         accion character(1) := 'D';
425         usuario varchar(30) := (select user);
426         fecha_registro date := now();
427         id_nulo integer := 0;
428         nombre_nulo character(1) := ' ';
429         id_tipo_habitat_nulo integer := 0;
430         id_continente_nulo integer := 0;
431
432
433     BEGIN
434
435         INSERT INTO public."BitacoraHabitat"
436             VALUES (OLD.id_habitat, OLD.nombre, OLD.id_tipo_habitat, OLD.id_continente,
437
438                     id_nulo, nombre_nulo, id_tipo_habitat_nulo, id_continente_nulo,
439
440                     usuario, accion, fecha_registro);
441
442         RETURN NEW;
443
444     END
445 $$
446 LANGUAGE PLPGSQL;
```



18.1 Desplegando la tabla habitat

```
858 select * from habitat;
859 DELETE FROM habitat where id_habitat = 6;
860 select * from "BitacoraHabitat";
861
```

Notifications		Data Output	Messages	Explain
	id_habitat [PK] integer	nombre character varying (30)	id_tipo_habitat integer	id_continente integer
1	336	sdknfsdknfjsdnf	10	255
2	337	sdfsdfsd	10	255
3	6	Tropical	10	255

18.2 Desplegando la bitacora habitat

```
858 select * from habitat;
859 DELETE FROM habitat where id_habitat = 6;
860 select * from "BitacoraHabitat";
861
```

Notifications		Data Output	Messages	Explain					
id_tipo_habitat_old integer	nombre_old character varying (30)	id_tipo_habitat_old integer	id_continente_old integer	id_habitat_new integer	nombre_new character varying (30)	id_tipo_habitat_new integer	id_continente_new integer	usuario character varying (15)	accion character (1)
336	sdfsdfsd	10	255	336	sdknfsdknfjsdnf	10	255	postgres	A
0		0	0	6	Tropical	10	255	postgres	I
6	Tropical	10	255	0		0	0	postgres	D





## 19 eliminar\_lista\_dezonas\_visitadas

```
480 -- eliminar lista zonas visitadas
481 CREATE FUNCTION eliminar_lista_dezonas_visitadas() RETURNS TRIGGER AS
482 $$
483     DECLARE
484
485         accion character(1) := 'D';
486         usuario varchar(30) := (select user);
487         fecha_registro date := now();
488         id_nulo integer := 0;
489         codigo_nulo varchar(30) := ' ';
490         id_zonas_nulo integer := 0;
491
492
493     BEGIN
494
495         INSERT INTO public."BitacoraLista_dezonas_visitadas"
496             VALUES (OLD.id_tours, OLD.codigo, OLD.id_zona,
497
498                     id_nulo, codigo_nulo, id_zonas_nulo,
499
500                     usuario, accion, fecha_registro);
501
502     RETURN NEW;
503
504     END
505 $$
506 LANGUAGE PLPGSQL;
507
```

## 19.1 Desplegando la tabla lista de zonas visitadas

```
880 select * from lista_dezonas_visitadas;
881 DELETE FROM lista_dezonas_visitadas WHERE id_tours = 6;
882 select * from "BitacoraLista_dezonas_visitadas;
883
884
885
```

Notifications   Data Output   Messages   Explain

	 id_tours [PK] integer 	codigo character varying (30) 	id_zona integer 
1	700	b737	5001
2	6	b737	5001

## 19.2 Desplegando la bitácora lista de zonas visitadas

```
880 select * from lista_dezonas_visitadas;
881 DELETE FROM lista_dezonas_visitadas WHERE id_tours = 6;
882 select * from "BitacoraLista_dezonas_visitadas";
883
```

Notifications Data Output Messages Explain

	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





## 20eliminar\_tipo\_habitat

```
539 -- eliminar tipo_habitat
540 CREATE FUNCTION eliminar_tipo_habitat() RETURNS TRIGGER AS
541 $$
542     DECLARE
543
544         accion character(1) := 'D';
545         usuario varchar(30) := (select user);
546         fecha_registro date := now();
547         id_nulo integer := 0;
548         clima_nulo character := ' ';
549         tipo_vegetacion_nulo character(1) = ' ';
550
551
552     BEGIN
553
554         INSERT INTO public."BitacoraTipo_habitat"
555             VALUES (OLD.id_tipo_habitat, OLD.clima, OLD.tipo_vegetacion,
556
557                     id_nulo, clima_nulo, tipo_vegetacion_nulo,
558
559                     usuario, accion, fecha_registro);
560
561     RETURN NEW;
562
563     END
564 $$
565 LANGUAGE PLPGSQL;
566
```

## 20.1 Desplegando la tabla tipo hábitat

```
896 select * from tipo_habitat;  
897 DELETE FROM tipo_habitat where id_tipo_habitat = 6;  
898 select * from "BitacoraTipo_habitat";  
899
```











Notifications Data Output Messages Explain

	 id_tipo_habitat [PK] integer 	clima character varying (30) 	tipo_vegetacion character varying (30) 
1	10	tropical	bcbovbcbcb
2	6	Soleado	Humedo

## 20.2 Desplegando la bitácora tipo habitat

```
896 select * from tipo_habitat;  
897 DELETE FROM tipo_habitat where id_tipo_habitat = 6;  
898 select * from "BitacoraTipo_habitat";  
899
```

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	bcbovbcbcb	postgres	A	2021-06-23
2	0			6	Soleado	Humedo	postgres	I	2021-06-25
3	6	Soleado	Humedo	0			postgres	D	2021-06-25

## 21 eliminar\_zona

```
54 -- eliminar zona
55 CREATE FUNCTION eliminar_zona() RETURNS TRIGGER AS
56 $$
57     DECLARE
58
59         accion character(1) := 'D';
60         usuario varchar(30) := (select user);
61         fecha_registro date := now();
62         id_nulo integer := 0;
63         nombre_nulo character(1) := ' ';
64         extension_territorial_nulo double precision := 0.0;
65
66     BEGIN
67
68         INSERT INTO public."BitacoraZona"
69             VALUES (OLD.id_zona, OLD.nombre, OLD.extension_territorial,
70
71                     id_nulo, nombre_nulo, extension_territorial_nulo,
72
73                     | usuario, accion, fecha_registro);
74
75     RETURN NEW;
76
77     END
78 $$
79 LANGUAGE PLPGSQL;
```

## 21.1 Desplegando la tabla zona

```
902 select * from zona;  
903 DELETE FROM zona where id_zona = 6;  
904 select * from "BitacoraZona";  
905
```

Notifications   Data Output   Messages   Explain

	id_zona [PK] integer	nombre character varying (30)	extension_territorial double precision
1	5001	Central	1555.2
2	6	Sur	506



## 21.2

## Desplegando la bitácora zona

```
901  
902 select * from zona;  
903 DELETE FROM zona where id_zona = 6;  
904 select * from "BitacoraZona";  
905
```

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















































