

# Escuela Politécnica Nacional

**Nombre:** Cristian Guamba

**Asignatura:** Bases de datos II

**Tema:** Proyecto Final

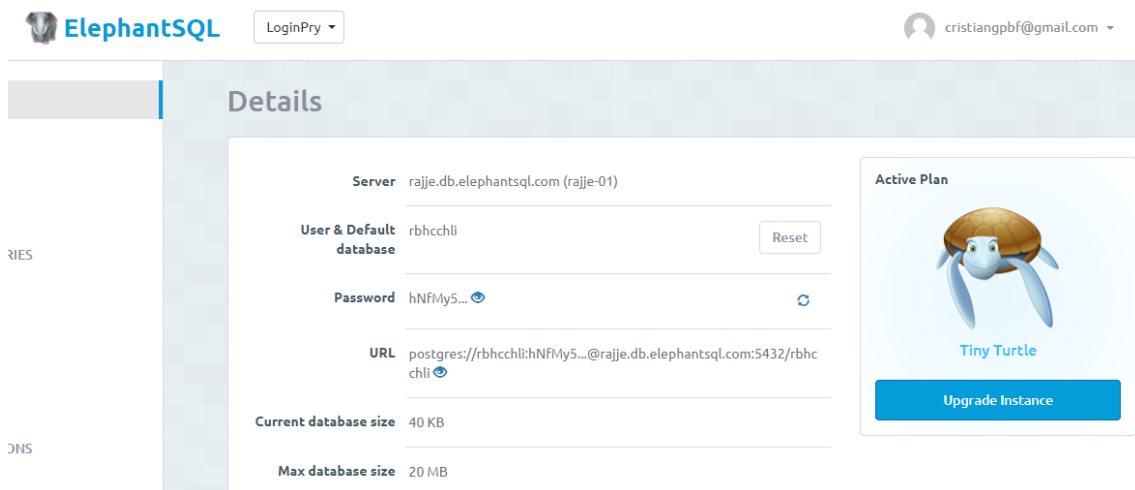
**Fecha de entrega:** 13/02/2020

**Creación de aplicación para registros de formularios 102A y 106 del SRI**

## Diseño de la base de datos

Para la conexión necesaria para el inicio de sesión se uso PostgreSQL

La base de datos se alojó en un servicio en línea para esta base de datos llamada elephant, la cual mediante una membresía gratuita, nos ofrece 20 Mb para almacenar bases de datos pequeñas.



Al realizar la conexión con el servidor mediante el administrador pgadmin, se procedió a crear las tablas respectivas y la inserción de usuarios de ejemplo.

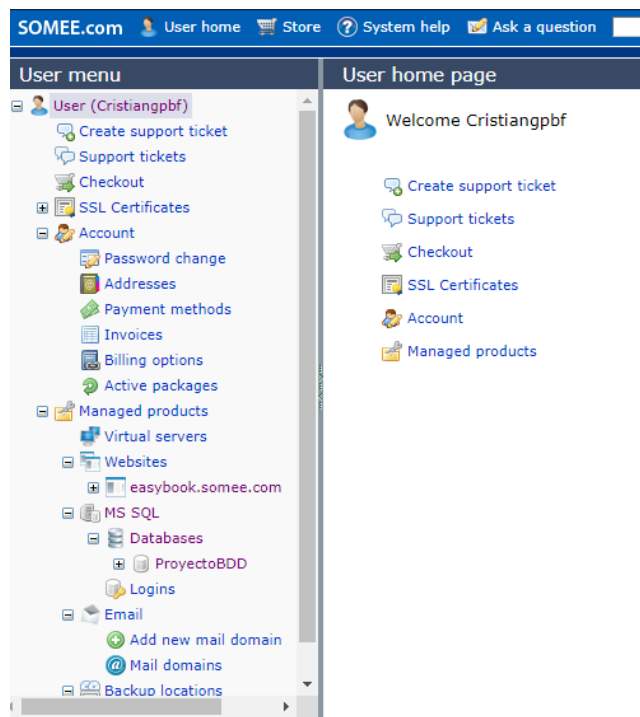
```
1 |-----Creación tabla usuarios-----
2 | create table tbl_user
3 | (
4 |     IDUSER INT GENERATED ALWAYS AS IDENTITY,
5 |     CEDULA character varying(13) not null,
6 |     PASS character varying(20) not null
7 | )
8 |
9 | select * from tbl_user
10 |
11 | insert into tbl_user (cedula,pass) values('1750024067','123456');
12 | insert into tbl_user (cedula,pass) values('1708009350','123456');
13 | insert into tbl_user (cedula,pass) values('1750024068','123456');
14 | insert into tbl_user (cedula,pass) values('1750024062','123456');
15 | insert into tbl_user (cedula,pass) values('1234567890','123456');
16 |
```

Luego se creó una función almacenada en la base de datos para validar si un usuario existe o no en la tabla de usuarios.

```
17 -----función para login-----
18 create function u_login(_cedula character varying, _pass character varying)
19 returns int as
20 $$
21 begin
22     if (select count(*) from tbl_user where cedula = _cedula and pass = _pass) > 0 then
23         return 1; -- Login satisfactorio
24     else
25         return 0;
26     end if;
27 end
28 $$
29 language plpgsql
```

Para el almacenamiento de los registros de los formularios 102A y 106 se usó la base de datos Sql server.

La base se alojó en la página somee, la cual ofrece un servidor compartido de SQLserver.



Al establecer una conexión mediante Sqlserver management studio, se procedió a crear las tablas para los registros de los formularios.

Las tablas se relacionan con los usuarios mediante los campos CED\_102A y CED\_106.

### Tabla para Auditorias

Para poder auditar las acciones realizadas sobre la base de datos de sqlserver se creó la siguiente tabla.

```

1 CREATE TABLE Audit(
2     AuditID [int]IDENTITY(1,1) NOT NULL,
3     Type char(1),
4     TableName varchar(128),
5     PrimaryKeyField varchar(1000),
6     PrimaryKeyValue varchar(1000),
7     FieldName varchar(128),
8     OldValue varchar(1000),
9     NewValue varchar(1000),
10    UpdateDate datetime DEFAULT (GetDate()),
11    UserName varchar(128)
12 )

```

La tabla de auditorias se alimenta mediante triggers que evalúan de manera detallada cada cambio realizado sobre la base de datos.

```

14 DECLARE @sql varchar(8000), @TABLE_NAME sysname;
15 DECLARE @TABLA_LIST varchar(8000) = ',REGISTRO_102A,REGISTRO_106,';
16
17 SET NOCOUNT ON
18
19 SELECT @TABLE_NAME= MIN(TABLE_NAME)
20 FROM INFORMATION_SCHEMA.Tables
21 WHERE TABLE_TYPE= 'BASE TABLE'
22 AND TABLE_NAME!= 'sysdiagrams'
23 AND TABLE_NAME!= 'Audit' AND CHARINDEX(','+ TABLE_NAME +',', @TABLA_LIST) > 0;
24
25 WHILE @TABLE_NAME IS NOT NULL
26 BEGIN
27
28 EXEC('IF OBJECT_ID (''' + @TABLE_NAME+ '_ChangeTracking'', ''TR'') IS NOT NULL DROP TRIGGER ' + @TABLE_NAME+ '_ChangeTracking')
29
30 SELECT @sql =
31 'create trigger ' + @TABLE_NAME+ '_ChangeTracking on ' + @TABLE_NAME+ ' for insert, update, delete
32 as
33 declare @bit int, @field int,@maxfield int,@char int,@fieldname varchar(128),@TableName varchar(128),
34 @PKCols varchar(1000),@sql varchar(2000),@UpdateDate varchar(21),@UserName varchar(128),@Type char(1),
35 @PKFieldSelect varchar(1000),@PKValueSelect varchar(1000)
36
37 select @TableName = ''' + @TABLE_NAME+ '''
38
39 -- date and user
40 select @UserName = system_user,
41 @UpdateDate = convert(varchar(8), getdate(), 112) + ' ' + convert(varchar(12), getdate(), 114)
42
43 -- Action
44 if exists (select * from inserted)
45 if exists (select * from deleted)

```

**Conexiones de la aplicación de escritorio con las bases de datos.**

## PostgreSQL

La conexión se realizó mediante el aditivo Npgsql, el cual permite realizar la conexión e interactuar con la base de datos.

```

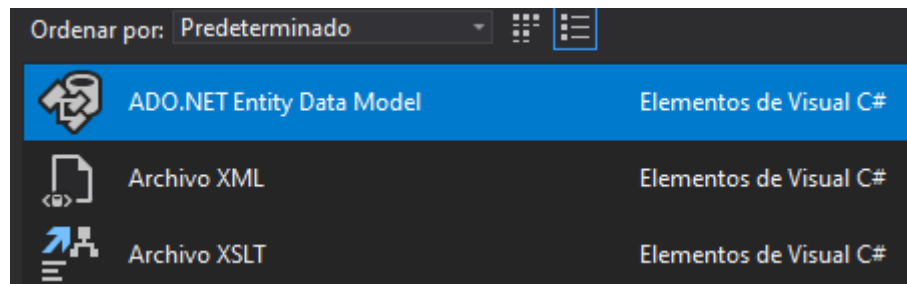
//Conexion con la base de datos mediante Npgsql

private NpgsqlConnection conn;
string connstring = String.Format("Server={0}; " +
    "User Id={1};Password={2};Database={3};",
    "rajje.db.elephantsql.com", "rbhcchli", "hNfMy507K3dGerBGkgVWptX08lrW27t8", "rbhcchli");
//private DataTable dt;
private NpgsqlCommand cmd;
private string sql = null;

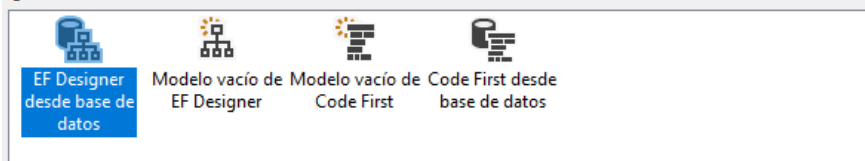
```

## SqlServer

Para SqlServer se usó el asistente de Visual Studio que permite realizar modelos de datos a partir de las tablas que ya existen en la base de datos. Mediante Entity Framework nos permite interactuar con la base de datos.



¿Qué debería contener el modelo?



Crea un modelo en EF Designer a partir de una base de datos existente. Puede elegir la conexión de base de datos, la configuración del modelo y los objetos de base de datos que se incluirán en el modelo. Las clases con las que interactuará la aplicación se generan a partir del modelo.

Asistente para Entity Data Model



Elegir la conexión de datos

¿Qué conexión de datos debe usar la aplicación para conectarse a la base de datos?

vb4800.ProyectoBDD.dbo

Nueva conexión...

Esta cadena de conexión parece contener datos confidenciales (por ejemplo, una contraseña) que son necesarios para conectarse con la base de datos. Almacenar datos confidenciales en la cadena de conexión puede suponer un riesgo para la seguridad. ¿Desea incluir estos datos en la cadena de conexión?

- ☐ No, excluir datos confidenciales de la cadena de conexión. Los estableceré en el código de mi aplicación.
- ☐ Sí, incluir datos confidenciales en la cadena de conexión.

Cadena de conexión:

```
metadata=res://*/models.Model1.csdl|res://*/models.Model1.ssdl|
res://*/models.Model1.msl;provider=System.Data.SqlClient;provider connection string="data
source=ProyectoBDD.mssql.somee.com;initial catalog=ProyectoBDD;persist security info=True;user
id=Cristiangpbf_SQLLogin_1;password=*****;MultipleActiveResultSets=True;App=EntityFramework
"
```

☒ Guardar configuración de conexión en App.Config como:

ProyectoBDDEntities3