Fecha: 06/09/12

Índice

[**1.** **Instalación de Postgesql y Postgis** 1](#_Toc334610777)

[**2.** **Instalación de la Solinette©** 2](#_Toc334610778)

[**3.** **Importación de los datos a geolocalizar** 10](#_Toc334610779)

[**4.** **Usar la Solinette**© 11](#_Toc334610780)

1. **Instalación de Postgesql y Postgis**

La instalación de Postgresql se hizó con la versión 9.1.5. El instalador esta en el compartido: [\\Compartido\sig\Processos y herramientas\Instaladores\SIG\postgresql-9.1.5-1-windows.exe](file:///\\Compartido\sig\Processos%20y%20herramientas\Instaladores\SIG\postgresql-9.1.5-1-windows.exe)

Este instalador incluye la instalación de PgAdmin3.

Escogí como contraseña para Postgresql la palabra “**pacivur**”.

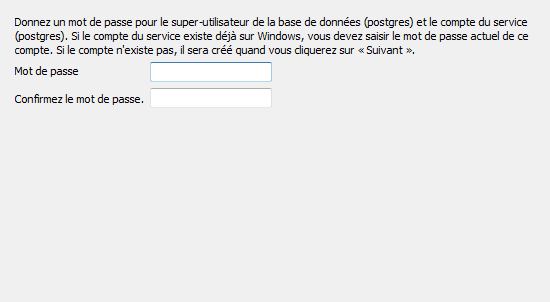


Figura 1: Contraseña

El número de puerto es el por defecto (5432)

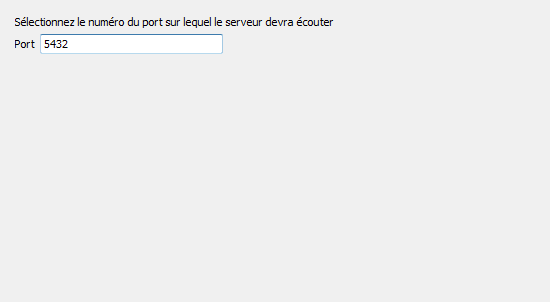


Figura 2: Número del puerto

La instalación de Posgis (versión 1.5) se hizó con el Stack Builder de Postgresql. Se debe parametrar el proxy (10.0.2.1, puerto: 3128).

Se usó la versión 1.5 de Postgis porque hay un bug (o un cambio) en la versión 2.0 que hace que la Solinette© no funciona.

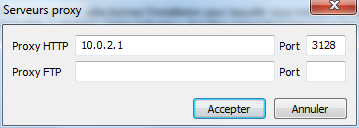


Figura 3: Parametros del proxy

1. **Instalación de la Solinette©**
2. Tenemos que crear una base de datos espacial. La llamamos “geolocalizacion”. No olvidarse de tomar, en la pestaña “Definición”, como modelo la base de datos espacial creada durante la instalación de Postgis (en nuestro caso se llama “template\_postgis”).

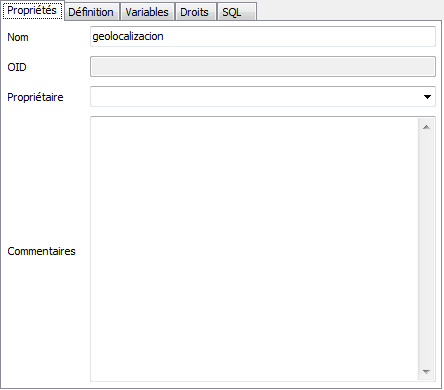


Figura 4: Creación de una nueva base de datos

1. Importamos la capa de vías en Postgis. La capa se llama “nombrevial\_09\_30.shp” y se encuentra en \\Compartido\sig\Processos y herramientas\solinette. La importación se hace con el plugin SPIT de QGIS. No olvidarse de poner el SRID a 32718 (UTM zona 18 Sur). La Solinette©, funciona con datos proyectados en EPSG:32718 (tanto para las vías que para las direcciones a ubicar).

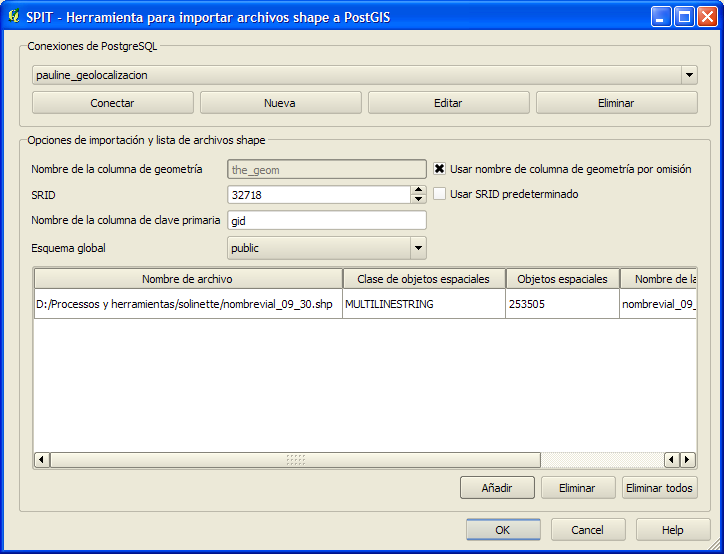


Figura 5: Importación de un shape en Postgis

Es necesario agregar un campo “ubigeo2”. Normalmente hay que hacer una intersección entre las vías y los distritos para llenar este campo. Aca solamente le doy el mismo valor que el campo “UBIGEO” asi que no sirve pero como esta en el algoritmo, lo dejo. A mejorar.

ALTER TABLE nombrevial\_09\_30 ADD COLUMN ubigeo2 char(6);

UPDATE nombrevial\_09\_30 SET ubigeo2 = "UBIGEO";

1. A partir de la capa de las vías importada, creamos una capa que será usada en la Solinette©. En Postgresql, ejecutamos la consulta siguiente:

-- edicion de la tabla de las vías

-- necesito hacer un un group by sobre varios campos y un st\_union(the\_geom) para no tener varios

-- objetos que tienen el mismo nombre, cuadra, tipo en el mismo distrito (pero solamente para los que tienen estos campos llenos)

create table nombrevial\_09\_30\_solinette as

(

select "CATEG\_VIA", "NOMBRE\_VIA", "NOMBRE\_ALT", "CUADRA", "IZQESQUEMA", "DERESQUEMA", "UBIGEO", ubigeo2,

st\_linemerge(st\_union(the\_geom)) as the\_geom from nombrevial\_09\_30

where "CATEG\_VIA" <> '' and "NOMBRE\_VIA" <> '' and "CUADRA" <> '' and "IZQESQUEMA" is not NULL and "DERESQUEMA"is not NULL

and "UBIGEO" <> '' and ubigeo2 <> ''

group by "CATEG\_VIA", "NOMBRE\_VIA", "NOMBRE\_ALT", "CUADRA", "IZQESQUEMA", "DERESQUEMA", "UBIGEO", ubigeo2

);

Luego agregamos un campo id a la nueva tabla. Llamamos este campo “gid”.

-- Agrego un campo gid

alter table nombrevial\_09\_30\_solinette add column gid int;

CREATE temp SEQUENCE seq\_id\_vias START 1;

update nombrevial\_09\_30\_solinette set gid = nextval('seq\_id\_vias');

**Es importante notar que si se modifica la capa inicial de las vías o si se usa una otra capa, hay que volver a ejecutar estas 2 consultas.**

1. A este nivel, tenemos una capa de vías lista para usar en la Solinette©. Ahora tenemos que importar la capa de distritos. Se llama “distritos\_bd.shp” y se encuentra en el compartido (\\Compartido\sig\Processos y herramientas\solinette). Para la importación hacemos como para la capa de vías.
2. Tenemos que tomar en cuenta las diferentes maneras de llamar un distrito. Por eso creamos un nuevo campo "NOMBRE2"

ALTER TABLE distritos\_bd ADD COLUMN "NOMBRE2" char(80);

UPDATE distritos\_bd SET "NOMBRE2" = "NOMBRE";

Ahora modificamos los nombres de distritos que son conocidos de diferentes maneras o que son a menudo escritos con errores ortográficos:

UPDATE distritos\_bd SET "NOMBRE2" = 'CERCADO DE LIMA' where "NOMBRE" = 'LIMA';

UPDATE distritos\_bd SET "NOMBRE2" = 'EL RIMAC' where "NOMBRE" = 'RIMAC';

UPDATE distritos\_bd SET "NOMBRE2" = 'CARMEN DE LA LEGUA' where "NOMBRE" = 'CARMEN DE LA LEGUA REYNOSO';

UPDATE distritos\_bd SET "NOMBRE2" = 'BREÑA' where "NOMBRE" = 'BRENA';

UPDATE distritos\_bd SET "NOMBRE2" = 'SURCO' where "NOMBRE" = 'SANTIAGO DE SURCO';

UPDATE distritos\_bd SET "NOMBRE2" = 'MAGDALENA VIEJA' where "NOMBRE" = 'PUEBLO LIBRE';

UPDATE distritos\_bd SET "NOMBRE2" = 'ATE' where "NOMBRE" = 'ATE VITARTE';

UPDATE distritos\_bd SET "NOMBRE2" = 'VILLA MARIA' where "NOMBRE" = 'VILLA MARIA DEL TRIUNFO';

UPDATE distritos\_bd SET "NOMBRE2" = 'EL CALLAO' where "NOMBRE" = 'CALLAO';

UPDATE distritos\_bd SET "NOMBRE2" = 'MAGDALENA' where "NOMBRE" = 'MAGDALENA DEL MAR';

UPDATE distritos\_bd SET "NOMBRE2" = 'SANTA MARIA' where "NOMBRE" = 'SANTA MARIA DEL MAR';

UPDATE distritos\_bd SET "NOMBRE2" = 'CHORILLOS' where "NOMBRE" = 'CHORRILLOS';

UPDATE distritos\_bd SET "NOMBRE2" = 'BARANCO' where "NOMBRE" = 'BARRANCO';

UPDATE distritos\_bd SET "NOMBRE2" = 'SAN MARTIN DE PORRAS' where "NOMBRE" = 'SAN MARTIN DE PORRES';

A este nivel tenemos todos los datos de base listos para usar la solinette.

1. Ahora necesitamos instalar el modulo python pscycopg2 para poder usar la solinette. Este modulo permite conectarse a Postgresql en Python. El instalador se encuentra [aquí](http://www.stickpeople.com/projects/python/win-psycopg/). Hay que escoger un instalador en función de sus versiones de Postgresql y Python. En nuestro caso escogimos el instalador **psycopg2-2.4.5.win32-py2.6-pg9.1.3-release.exe**
2. Para poder usar la Solinette© en una red hay que configurar Postgresql. Primero hay que editar el archivo pg\_hba.conf (que se encuentra en C:\Program Files\PostgreSQL\9.1\data) y al final del archivo, debajo de la línea

host all all 127.0.0.1/32 md5

Hay que agregar la línea (si queremos permitir la conexión de cualquier dirección IP)

host all all 0.0.0.0/0 md5

Después se debe reiniciar el servidor Postgresql para que tome en cuenta la nueva configuración.

Luego tenemos que permitir las conexiones a Postgresql en nuestra maquina a través del Firewall. En Windows 7 hay que seguir las instrucciones [siguientes](http://windows.microsoft.com/es-ES/windows7/Open-a-port-in-Windows-Firewall).

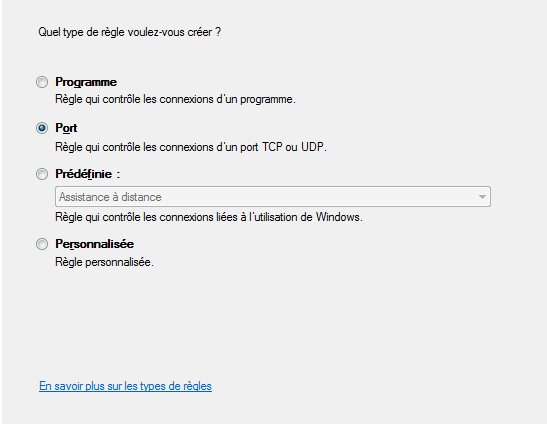


Figura 6: El tipo de regla es Puerto

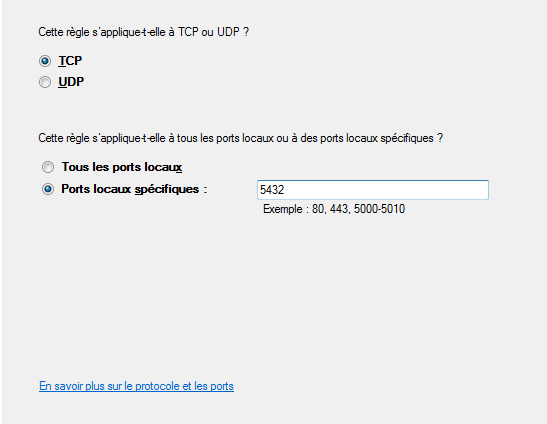


Figura 7: El número de puerto es 5432

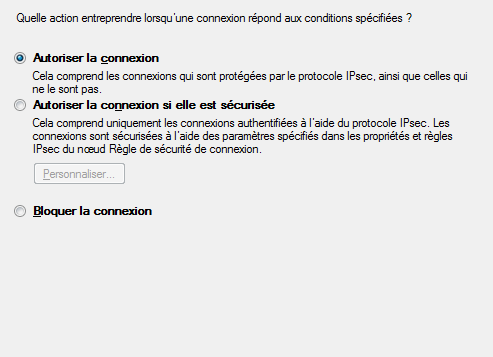


Figura 8: Autorizamos la connexión a través de este puerto

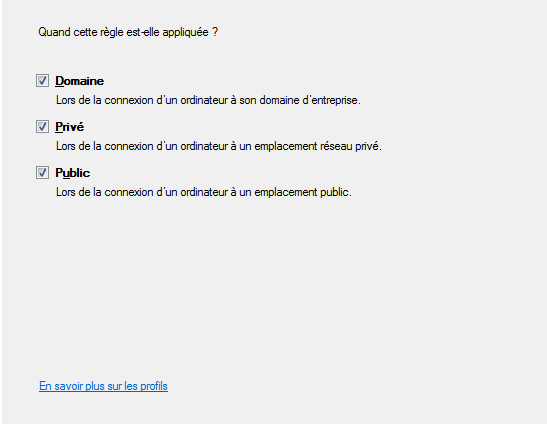


Figura 9: Aplicamos esta regla a todos los casos

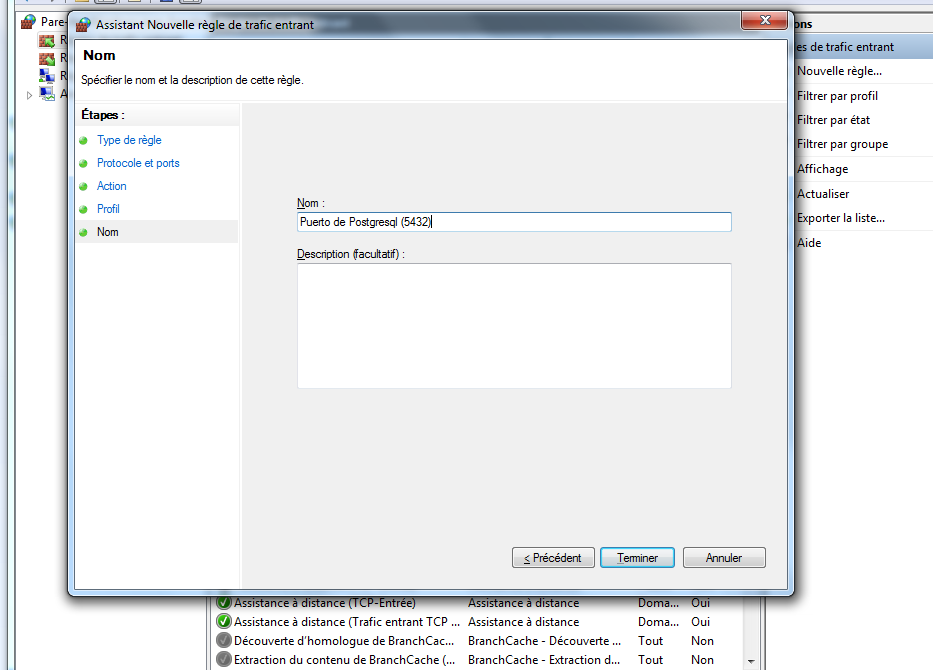


Figura 10: Le damos un nombre

1. **Importación de los datos a geolocalizar**

Antes de geolocalizar las direcciones con la Solinette©, hay que importar estos datos en una tabla PostgreSQL.

Primero debemos convertir nuestros datos al formato .TXT o .CSV. Luego crear una nueva tabla en PostgreSQL. Esta tabla albergará nuestros datos. A continuación, un ejemplo de creación de una nueva tabla:

create table empresas\_mml (

id int,

nombre char(100),

direccion char(100),

distrito char(50),

telefono char(40),

email char(100)

);

Los campos corresponden a los campos del archivo que contiene los datos que queremos importar. Para funcionar, la Solinette© necesita al 3 campos:

* Un campo que contenga un identificador único (de tipo entero) con valores que se incrementan de uno en uno, no es necesario que empiece en 1 (entonces puede ser 1, 2, 3, etc… o 37, 38, 39, etc…)
* Un campo que contenga la dirección, con el tipo de la vía, el nombre y el número de puerta. Puede también contener complementos de dirección (piso, AH, urbanización, etc…) pero no será tomado en cuenta.
* Un campo que contenga el nombre del distrito

No hay ninguna obligación sobre la denominación de estos 3 campos (estos 3 campos deben ser mencionado en el script python). El orden de los campos en la tabla tampoco importa.

La tabla de direcciones no puede contener campos que se llamen ‘tipo\_via’, ‘nombre\_via’, ‘numero’, ‘complemento\_dir’ o ‘ubigeo’ porque estos campos son creados durante la ejecución de la Solinette©.

Una vez que tenemos una tabla (vacía) podemos importar los datos a partir de nuestro archivo .TXT o .CSV.

A continuación la consulta para importar los datos desde un archivo .CSV en la tabla que acabamos de crear:

SET CLIENT\_ENCODING TO 'LATIN1';

copy empresas\_mml from '\\Compartido\sig\Processos y herramientas\solinette\datos a ubicar\empresas\_MML.csv' DELIMITER ';' CSV HEADER;

1. **Usar la Solinette**©

Antes de usar la Solinette© tenemos que editar el script “**solinette\_1.4.1.py**” para que tome en cuenta nuestros datos (Con una interfaz gráfica eso ya no será necesario).

Pero antes tenemos que parametrar la conexión a Postgresql. Eso se hace alrededor de la línea 265:

conn = psycopg2.connect('host=localhost port=5432 dbname=geolocalizacion user=postgres password=pacivur')

Los mismos parámetros se deben editar alrededor de la línea 1285:

conn2 = psycopg2.connect('host=localhost port=5432 dbname=geolocalizacion user=postgres password=pacivur')

Los parámetros (de las tablas de direcciones y de vías) a cambiar están alrededor de la línea 275.

tabla\_direcciones = 'empresas\_mml'

tabla\_vias = 'nombrevial\_09\_30\_solinette'

…

…

# columnas de la tabla de las direcciones

col\_id = 'id'

col\_direccion = 'direccion'

col\_dist = 'distrito'

En la primera línea tenemos que indicar el nombre de la tabal que contiene las direcciones (en nuestro ejemplo es la tabla “empresas\_mml”).

En la segunda línea, tenemos que indicar el nombre de la tabla de las vías (“nombrevial\_09\_30\_solinette”).

Las variables” col\_id”, “col\_direccion” y “col\_dist” deben tenr como valores, respectivamente el nombre de las columnas que contienen el id, la dirección y el distrito dentro de nuestra tabal de datos a ubicar (en nuestro ejemplo “empresas\_mml”).

Una vez el script editado, lo podemos ejecutar en Python. Abrimos el script en el IDE Python y lo ejecutamos presionando la tecla *F5*.

En la figura 11, a continuación, se ve el resultado de la ejecución de la Solinette© en la consola Python.

Hay 4 etapas dentro del programa:

1. El “parsing” de la dirección en varios campos para que sea posible encontrar la cuadra correspondiente en la tabla de las vías.
2. Afectación de un código UBIGEO a la dirección en función de su nombre de distrito
3. Localización de las direcciones (búsqueda de la cuadra correspondiente en la tabla de las vías e interpolación en función del número de la dirección.
4. Creación de las tablas finales

La Solinette© crea 4 nuevas tablas con los sufijos siguientes (el cuerpo del nombre de estas 4 tablas es el mismo que el nombre de la tabla que contiene las direcciones a ubicar, es decir en nuestro ejemplo, “empresas\_mml”):

* **\_geom**: contiene las direcciones geolocalizadas. Esta tabla contiene un campo de tipo *geometry,* entonces puede ser visualizada en Postgis y luego exportado al formato *shapefile*.
* **\_multi**: contiene las direcciones para las cuales se han encontrado más de una cuadra idéntica (es decir, mismo nombre de vía, mismo número de cuadra y mismo distrito). En este caso la Solinette© no sabe a qué cuadra pertenece realmente la dirección y no la geo ocaliza. Una solución para poder geolocalizar estas direcciones es de editar la tabla de las vías y volver a ejecutar la Solinette© sobre esta tabla.
* **\_bug**: Las direcciones que tienen un formato correcto (nombre de vía, número de puerta y nombre de distrito) pero para las cuales ne se encuentra una cuadra correspondiente en la tabla de las vías.
* **\_imposible**: Las direcciones que no tienen un nombre de vía y/o un número de puerta. En este caso, la Solinette© no puede intentar localizarlas.

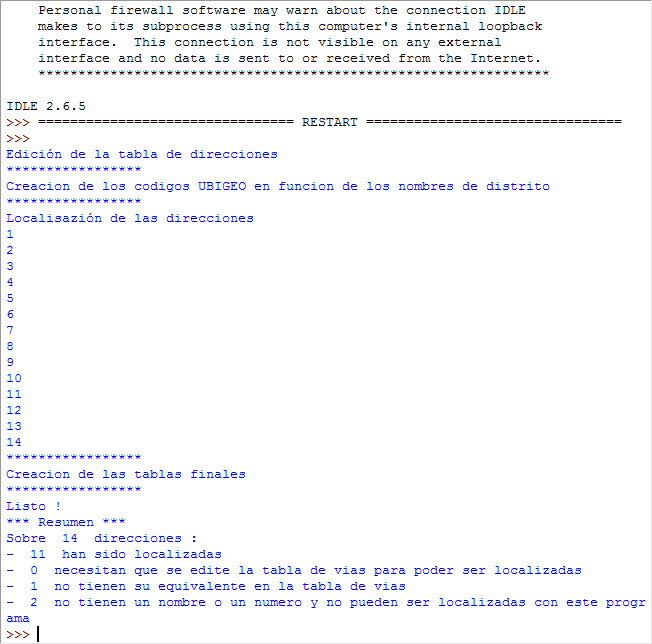


Figura 11 Resultado de la Solinette

Además de crear nuevas tablas, la Solinette© edita la tabla inicial de direcciones (en nuestro ejemplo, “empresas\_mml”), agregándole los campos siguientes:

* **tipo\_via** (tipo texto) : Un código que corresponde al tipo de vía (avenida, calle, jirón, etc..)
* **nombre\_via** (tipo texto): El nombre de la avenida (en mayúsculas y sin accentos)
* **numero** (tipo numérico entero): El número de la dirección
* **complemento\_dir** (tipo texto): El complemento de la dirección, si hay uno. Puede ser un número de piso, una urbanización, un AH.
* **ubigeo** (tipo texto): El código UBIGEO del distrito en el cual se encuentra la dirección.