

# Explosión de Puntos en Manzanas - Scripting

Caracterización de problemas en tecnologías SIG en scripting y abordaje de los mismos - Caso de estudio

## Resumen:

En el contexto del Correo Uruguayo, es de vital interés el georeferenciado de las direcciones postales de Uruguay. Del censo de 2011 a cargo del INE se obtuvo las direcciones de todo el Uruguay, aunque no fueron espacializadas. Un proceso posterior pudo ubicar cada dirección en el centroide de su zona censal (en general la manzana) correspondiente. Éste, es el punto inicial del problema. El objetivo, por lo tanto es llevar 1,5 millones de puntos (direcciones) a la cara de la zona (en general la cuadra) a la que pertenecen y distribuirlos según el número de secuencia que poseen. La solución se consiguió abstrayendo definiciones tales como “cuadra”, “calle”, “manzana”, y estableciendo cada elemento a través de sus propiedades. El enfoque tiene una visión híbrida, aplicando conocimientos de programación generales y conceptos y técnicas del mundo SIG.

Se utilizó como herramienta el módulo de scripting de gvSIG Desktop 2.2.

Este artículo resuelve el problema planteado pero también establece y estudia los impedimentos que surgen en el abordaje y la resolución.

No hay piezas de código explícitas.

Palabras claves: Scripting, gvSIG, problema, direcciones, georeferenciar.

Fabián Coscia.

Técnico SIG, Unidad de Geomática, Correo Uruguayo, Uruguay.

fabiancoscia@gmail.com.

## 1. Introducción

La Unidad de Geomática del Correo Uruguayo se ocupa fuertemente de tareas que hacen a la geo-localización de direcciones postales. Es así que en el marco del censo de 2011, llevado adelante por el INE, se obtuvo una actualización de la cartografía y nomenclátor. Sin embargo las direcciones que se relevaron no fueron espacializadas, es decir que no tienen geometría asociada, aunque sí guardan referencia a la zona censal<sup>1</sup> donde deberían ubicarse. Las zonas censales cuentan con geometría. Primero se estableció un proceso -que no cubre este artículo- que toma cada dirección y le asigna una geometría determinada, para esto se valió de la relación antes nombrada<sup>2</sup>. Como consecuencia se obtuvo un shapefile<sup>3</sup> de puntos, donde cada punto es una dirección, y su ubicación es el centroide de cada zona censal a la que pertenece.

### 1.1. Motivación

La motivación está dada por obtener una base de direcciones del INE geo-referenciada de tal forma que cada punto esté ubicado donde corresponda<sup>4</sup>, ya que se trata de información validada en campo y por tanto tiene un valor superlativo para actualizar o cotejar la base de datos de direcciones del Correo Uruguayo.

---

<sup>1</sup> El INE ha dividido el territorio uruguayo con fines estadísticos en porciones. La zona censal es la unidad menor identificable. En localidades censales o áreas amanzanadas generalmente coincide con una manzana.

<sup>2</sup> La relación está definida en la tabla de direcciones donde cada una tiene una referencia a qué zona censal pertenece.

<sup>3</sup> El formato ESRI Shapefile (shp) es un formato de archivo informático propietario de datos espaciales desarrollado por la compañía ESRI.

<sup>4</sup> Se manejan márgenes de error, valorando éstos y sobre todo su casual.

## 1.2. Objetivos

El objetivo de este artículo es establecer y caracterizar los problemas y soluciones que surgen al intentar resolver casos de estudio con características espaciales similares al presentado. Además, interesa llegar a una solución concreta que resuelva el problema y permita estudiar el potencial, las ventajas y desventajas de los módulos de scripting de herramientas SIG de software libre.

Cabe acotar que la solución al caso de estudio está dada por llevar cada punto (dirección) a su ubicación espacial correspondiente, esto es a las coordenadas de la puerta de calle del mismo, se manejan márgenes de error que se introducen en su momento.

## 2. Herramienta - Software seleccionado

En esta sección se evalúa la adopción de una herramienta de software que ayude a cumplir los objetivos en función de prioridades y necesidades que se describen.

El volumen de datos a procesar es de aproximadamente 1.5 millones de puntos, por lo que la lógica indica que la solución debe ser una tarea automatizada. Además sabemos que un script es un conjunto de instrucciones generalmente almacenadas en un archivo que deben ser interpretados línea a línea en tiempo real para su ejecución y que permite la automatización de tareas. Por lo que la búsqueda está enfocada en torno a herramientas SIG con módulos de scripting. También es prioridad la utilización de software libre.

La decisión del software a adoptar fue tomada en base a experimentación con gvSIG y un competidor directo como es QGIS y su módulo de scripting. Se hicieron pruebas de modificación de atributos en un volumen grande de datos, con una y otra herramienta. El resultado arrojó que QGIS resulta mucho mas performante, robusto y estable. Sin embargo, la curva de aprendizaje de PyQGIS, el lenguaje de programación de scripting de QGIS, es muy elevada frente a la sencillez que ofrece Jython, el lenguaje de gvSIG. Este aspecto es determinante a la hora de atacar un problema de cierta complejidad, ya que la herramienta a utilizar no debe aumentar la dificultad, sino lo contrario, por lo que el software utilizado es gvSIG Desktop 2.2<sup>5</sup>, y en particular su módulo de scripting. El lenguaje de desarrollo (Jython) está basado Python y algo de Java.

## 3. Hipótesis iniciales

Llamaremos hipótesis iniciales a los datos que se poseen y se emplean para cumplir el objetivo. A continuación se listan y adjunta una breve descripción de ellos y sus características sobresalientes. Cabe acotar que solo se mencionan los atributos de cada capa que son de interés al artículo.

*c\_ejes\_calles*: capa de líneas que representa las calles de Uruguay. Esta capa posee el atributo *nombre\_11* que contiene el nombre de cada entidad<sup>6</sup> calle.

*c\_zonas\_censales*: capa de polígonos que representa las zonas censales del país. Esta capa posee un atributo identificador.

*c\_puntos*: capa de puntos que representa los domicilios de todo el país. Esta capa posee el atributo *nombre\_cal*: nombre de la calle a la que pertenece el punto,

---

<sup>5</sup> La versión mas estable al momento de este proyecto.

<sup>6</sup> Llamaremos entidad a un elemento de una capa. En el caso de una capa de líneas, cada línea es una entidad distinta.

*número*: número de puerta del domicilio que representa, *zona\_c*<sup>7</sup>: identificador de la zona censal a la que pertenece y *numero\_dom*: número secuencial asignado a cada domicilio y que establece un orden entre puntos dentro de una zona censal, recorriendo esta por su perímetro. Por ejemplo: el primer punto<sup>8</sup> de la zona tiene *numero\_dom* = 1, el siguiente punto *numero\_dom* = 2 y así sucesivamente hasta recorrer el perímetro de una zona. Es así que el punto anterior al primero<sup>9</sup>, posee el atributo *numero\_dom* = (total de puntos de la zona) ya que cierra el perímetro de la zona censal. Además cuenta con un atributo que identifica cada entidad.

En cuanto a ubicaciones espaciales, podemos considerar que entre zonas censales contiguas al menos hay un eje de calle en una ubicación aproximada. La noción de aproximado deja abierta varias posibilidades, como ser que el eje esté completamente contenido en una zona y no intercepte su lindera, que el eje zigzaguee o simplemente cruce de una zona a otra o que coincida entre el límite de contacto entre dichas zonas. Esto se puede observar en la Figura 1, en (a), (b) y (c) respectivamente.

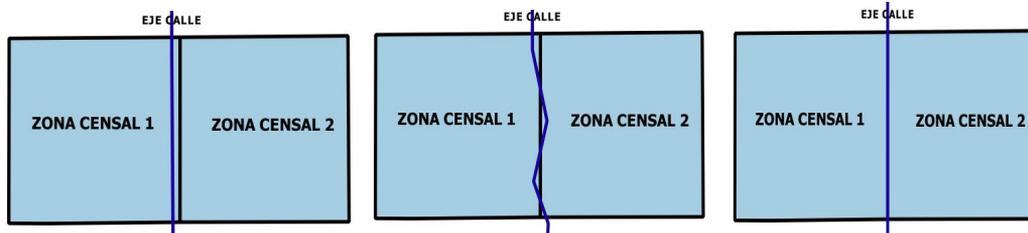


Figura 1 – (a), (b) y (c).

Se asume que la totalidad de la superficie a tratar de la República Oriental del Uruguay está particionada en zonas censales en la misma capa, donde no hay “huecos” sin zonas. Cabe acotar que no se poseen mas datos de utilidad que los expuestos.

#### 4. El problema

Como se expuso, el problema se plantea con zonas censales y ejes de calles con geometrías correctas, y direcciones ubicadas en el centroide de la zona a la que pertenecen. El problema se dará como resuelto si se llevan los puntos a donde geográficamente pertenecen con una tolerancia de cierto margen de error.

En la Figura 2 (a), se puede ver el planteo del problema, mientras que en la Figura 2 (b), la solución esperada.

<sup>7</sup> Este atributo no existe, se lo incluye en el documento para facilitar la presentación ya que es un valor calculado de otros atributos.

<sup>8</sup> El primer punto: es un punto arbitrario del comienzo de una cuadra de los lados de una zona censal.

<sup>9</sup> Asumiendo el orden explicado.



Figura 2 - planteo (a), solución (b)

Dado las hipótesis iniciales, un primer razonamiento ligero podría ser: tomando una única zona censal y teniendo el nombre de calle al cual pertenece cada punto, se los puede ubicar en la cuadra de la zona censal que se corresponde con la calle. Luego se podrían ordenar con el número de domicilio. Luego, repetir el razonamiento con el resto de las cuadras de la zona censal y luego repetirse con el resto de las zonas de la capa de zonas. Un primer inconveniente a este razonamiento es la definición espacial de cuadra. Informalmente, se podría intentar definir como el perímetro de la manzana que está ubicado entre dos esquinas contiguas. Esta definición introduce una nueva ambigüedad: la definición de esquina. Una posible definición es a través de las características de la línea que la define. Ya sea un eje de calle o el perímetro de la zona, una línea se construye como la unión de un conjunto de puntos, estos no siempre están alineados (no son líneas rectas). Un posible criterio puede ser definir esquina a partir de un determinado valor del ángulo de quiebre (vértice) de la línea ( $\alpha$  en la Figura 3).



Figura 3 – ejemplos de esquina (a), (b),(c),(d)

En la Figura 3, podemos ver las distintas formas que puede adoptar una línea a partir del valor del ángulo  $\alpha$ . Para cualquiera de los casos (a),(b),(c) o (d) existen ejemplos de esquinas y contraejemplos. A continuación se lista un contraejemplo de esquina para cada configuración de  $\alpha$ . Ya que los ejemplos son el común de los casos no se los menciona. Las calles son tomados del departamento de Montevideo, Uruguay.

- Para (a): Francisco de Sostoa es un contraejemplo de esquina.
- Para (b y c): Yi entre La Paz y Valparaíso son contraejemplos de esquina.
- Para (d): Cerrito entre Guaraní y Juan Lindolfo Cuestas

Se puede concluir que el ángulo del vértice que define una calle (una línea) no define de forma tácita si se está frente a una esquina o no. Por lo que no podemos considerar el ángulo con tal fin. - Cabe destacar que la idea de definir esquina era con el mero fin de esclarecer el concepto de cuadra de una zona censal-.

Se podría definir cuadra (de la zona) siguiendo otros mecanismos, como ser con la intersección de calles o estrategias similares pero llegaríamos a otras situaciones de ambigüedad, como por ejemplo por haber varias intersecciones entre los mismos ejes de calles, etc.

Este es un típico problema que se desea establecer, estudiar y dar solución en este artículo. Problemas donde los datos guardan una relación espacio-visual y por este motivo se hace sencillo efectuar un razonamiento apoyado en dicho componente, sin embargo, no todas las características espaciales, o visuales se pueden mapear de manera fidedigna a un script, un programa o simplemente a un razonamiento riguroso atado a datos y valores concretos.

El razonamiento inicial, que derivó en las definiciones de cuadra y esquina, solo intentaban ser una introducción a la dificultad del problema planteado a resolver y una presentación general del tipo de problemas que interesa trabajar en este artículo.

## 5. La solución

A continuación se plantea el enfoque de la solución y se procede con la misma. De aquí en mas solo se plantean y estudian los problemas que resulten de la solución adoptada.

El abordaje de la solución tiene un enfoque reduccionista, es decir que se identifica y resuelve cada problema por separado para luego unirlos en una solución completa. Otra característica que cabe resaltar es que se implementa a través de scripts, como se expuso en la *sección 2 Herramienta – Software seleccionado*.

### La manzana

El primer problema que se identifica es resolver en qué parte geográfica de la zona censal deben estar ubicados los puntos de las direcciones y hallarla. Independiente de las coordenadas específicas, se espera encontrar una zona que los delimite.

Se establece como una solución ideal que a los puntos de las direcciones dentro de una zona censal, se les asigne la misma ubicación de la puerta de calle de cada domicilio al que corresponden. Para esto hay que entender que representa la zona censal. Según la definición del INE, generalmente una zona censal coincide con una manzana y su límite lo determina otra zona censal. Es decir que la capa de zonas censales es una partición de la superficie de Uruguay que no deja “huecos”, como se expuso en la *sección 3 Hipótesis iniciales*. Por lo que entonces una zona censal la comprende la manzana catastral<sup>10</sup> mas la acera y la calzada que rodea la manzana. Entonces, si tomamos un polígono de una zona censal, la parte que se desea encontrar es la manzana, y en particular el perímetro de esta, que es el perímetro de un polígono semejante a la zona y de menor proporción, concéntrico a la misma.

En la Figura 4 se observa en verde oscuro la zona censal y en verde claro una zona semejante de menor proporción concéntrica a la misma como ejemplo.



Figura 4 - zona censal y la manzana

La proporción entre la zona censal y su semejante está dada por la distancia entre un padrón de una manzana y el medio en la calzada. Dado que el ancho de las calles en todo el

<sup>10</sup> Definición manzana catastral.

Uruguay es muy variante, se estima esta distancia en 6 metros.

De aquí en mas al polígono calculado lo llamaremos manzana ya que es lo que representa.

Se determinó empíricamente donde se deben ubicar los puntos: en el perímetro de la manzana hallada. Para hallar la solución de forma práctica se crea un script que tome todas las zonas de la capa `c_zonas_censales`, y que le calcule un buffer de -6 metros, construyendo así, una capa nueva con cada manzana calculada.

No es necesario que la capa de manzanas sea realizada con un script, ya que la función buffer es típica dentro de aplicaciones SIG.

### Orden de los puntos

El problema inicial presentado, requiere de la ubicación precisa de cada punto de dirección, teniendo en cuenta el atributo número de domicilio, por lo que se entiende que hay un problema subyacente de orden entre puntos. Por lo tanto, el segundo problema que se identifica, es hallar el orden entre los puntos de cada zona censal. Dichos puntos están almacenados en la capa `c_puntos`.

Si se toma una zona censal, los puntos de direcciones se encuentran concentrados bajo una misma ubicación y sin ningún tipo de relación, al menos que se pueda asumir. Es decir que si tomo los puntos uno a uno de una zona censal, y luego se examinan sus atributos nada me garantiza que el primero tenga como atributo `numero_dom = 1` y/o que el segundo tenga `numero_dom = 2`, por ejemplo. Aquí no hay componentes geográficas que dificulten la operativa. Sin embargo y con poca rigurosidad, se puede analizar que el orden tiene una relación directa con la calle a la que pertenece cada punto. A su vez, puntos con distinto número de domicilio pero con igual número de puerta, deben compartir la ubicación geográfica, ya que esa configuración define la existencia de un edificio: varios domicilios con una puerta y de calle en común.

Para poder identificar cada entidad es vital manejar su identificador, por lo que este se agrega a las relaciones de orden entre puntos que se plantean.

Tomando en cuenta los factores anteriores, la noción de orden entre puntos incrementa su complejidad.

Para ordenar los puntos de tal manera que plasmen las relaciones anteriores se opta por crear tres estructuras de datos por zona censal.

- un diccionario de calles: a partir de una calle devuelve un nuevo diccionario, que a partir de un número de puerta devuelve los identificadores de puntos que se corresponden con dicho número.
- una lista ordenada de calles(según el número de domicilio de los puntos)
- un diccionario de calles: a partir de una calle devuelve una lista ordenada de números de puerta.

En un script se puede trabajar directamente con las entidades de una capa (en este caso puntos) o con sus atributos. La primera opción agrega dificultad e inestabilidad a la ejecución de la solución en contraposición a la segunda. En este artículo se sigue la segunda opción. Claro está que para trabajar con los atributos de las entidades primero hay que acceder a ellas para obtenerlos, sin embargo se trata de manipularlas lo menos posible, esto es para obtener los atributos de interés y luego para su modificación.

Para la construcción de las estructuras se crea un script que toma cada zona censal de la capa `c_zonas_censales` y obtiene las entidades de la capa `c_puntos` con un filtro: el identificador de la zona censal y además que la respuesta esté ordenado por el atributo `numero_dom` en forma ascendente. Luego se recorre elemento por elemento de la salida anterior y se pueblan las estructuras presentadas.

Aunque no sea clara la construcción y pertinencia de las estructuras anteriores son

suficientes para mantener las relaciones de orden planteadas entre cada uno de los componentes, además de ser funcionales a las necesidades que se presentan en la próxima sección.

Cabe acotar que el uso de diccionarios en el lenguaje de programación utilizado (Jython) mantiene la relación <clave:valor>, pero no mantiene el orden entre claves. Dicha desventaja se resuelve con redundancia en las mismas estructuras y el uso de listas que si mantienen el orden.

### Las calles y el lugar de los puntos

Hasta este momento, si se toma una zona censal, se ha definido en qué parte geográfica corresponde llevar los puntos de las direcciones que se ubican acumulados en el centroide de la zona, también se estableció un orden entre dichos puntos con estructuras funcionales a nuestras necesidades, ya que a partir de una calle puedo obtener qué puntos corresponden a ella<sup>11</sup> de forma ordenada por ejemplo. El siguiente problema identificado es, primero, hallar que calles circunvalan la zona censal y luego, se deben llevar los puntos que corresponden a las calles encontradas a su cuadra correspondiente. Ambos problemas los vamos a atacar en conjunto.

En la *sección 4 El problema* se introdujo uno de los problemas anteriores y un razonamiento de cómo llevar los puntos a la cuadra de la zona a la que corresponden sin resultado positivo, cabe destacar que se concluyó que se debe al componente espacio-visual que ofrece información rica en análisis pero difícil de interpretar como datos concretos, o que al menos no permite su caracterización en una definición. Este tipo de problema es transversal y de particular interés en el presente artículo.

La solución que se encuentra es abordar la problemática a través de propiedades espaciales y no con definiciones. Esto plantea un problema severo: no dar una metodología general para su resolución, ya que cada solución va a depender del problema, de los datos, las relaciones espaciales y “planas” entre ellos y del ingenio del responsable de su solución. Incluso, se agrega la incertidumbre de encontrar varias soluciones aproximadas y tener que evaluar cuál es mejor en términos geográficos, dicho de otra manera cuál tiene menos error.

Establecer una cota de error en la problemática consiste en un problema en sí, ya que al obtener una solución concreta hay que recorrer el camino inverso encontrando una relación geográfica que nos permita identificar si la solución está dentro de los márgenes aceptables o no y encontrar un valor que caracteriza dicho error.

Procediendo en orden, primero se haya qué calles circunvalan la zona censal y luego se lleva los puntos a las calles correspondientes<sup>12</sup>. Para obtener las entidades calles de la capa `c_ejes_calles` que circunvalan la zona censal hay dos enfoques:

1. a partir de los puntos: a través del atributo `nombre_cal` que tienen los puntos como atributo, y luego obtener las calles de la capa que coinciden su atributo `nombre_11` con `nombre_cal`.
2. a partir de las calles: hallar las entidades calles que circunvalan la zona censal a través de propiedades espaciales.

La opción 1. en principio denota simplicidad. Aunque un análisis mas profundo indica que si el nombre de calle de alguno de los puntos es incorrecto, produce a un error difícil de

---

<sup>11</sup> No se obtienen los puntos directamente sino sus identificadores.

<sup>12</sup> Se permite el abuso de esta expresión con fin esclarecedor, ya que los puntos no van ubicados propiamente en las calles.

estimar a esta altura de la solución. Pero sin dudas conduciría a un falso positivo.

Otras dificultades con el mismo destino, es la existencia de varias calles distintas que comparten nombre. Además, si un punto tiene un nombre de calle incorrecto es más probable encontrar una calle que se corresponda en nombre con el punto, que viceversa, ya que el volumen de calles es muy superior en comparación a los nombres de calles que pueden poseer los puntos de una zona censal.

Cabe acotar que igualmente al identificar una entidad calle, no se sabe dónde está ubicada con respecto a la zona y los puntos, por lo que igualmente habría que hacer algo como lo planteado en la opción 2 para resolverlo.

Por las desventajas anteriores se buscan propiedades espaciales que permitan tomar la opción 2. Entonces, debemos encontrar los ejes de calles que circunvalan una zona para poder ubicar los puntos en función de dichos ejes.

Tomando una zona censal, y dado que esta rodeada por otras linderas, y que entre una zona y otra pasa un eje de calle, se plantea que una perpendicular a la geometría intersección entre la zona y una de sus linderas (una recta, ya que solo se tocan) corta el eje de calle que pasa entre dichas zonas. Por lo que de la capa de calles, la que corta a dicha perpendicular será una de las calles que circunvala la zona tomada. Cabe acotar que la intersección de la perpendicular puede arrojar el corte con más de un eje. En ese caso se toma el eje tal que el punto intersección sea más cercano a la intersección de las zonas, tomado por la perpendicular.

Siendo un poco más detallista, existen casos donde entre una zona y otra, existe un eje de calle que termina y otro que empieza con diferentes nombres.

El razonamiento planteado solo detectaría un solo eje. Un paliativo que se adopta es el siguiente: dado que la intersección entre ambas zonas es una recta, y una recta está compuesta por puntos, se calcula un segmento inicial entre el primer punto de la recta y el siguiente. Por el punto medio de dicho segmento se traza una perpendicular, con lo que la perpendicular corta a un eje de calle que pasa por un extremo entre las zonas. De igual manera se razona con el otro extremo de la intersección de zonas, solo que esta vez se calcula un segmento entre el último punto de la intersección y el anterior. Nuevamente se traza una perpendicular por el punto medio y se obtiene una intersección entre la perpendicular y al menos una calle. En cada extremo de la intersección de zonas, se toma el eje más cercano a esta.

Es interesante ver que la recta intersección entre zonas puede tener solo dos puntos lo que haría coincidir el segmento inicial y final y también sus perpendiculares hallando un solo eje.

Otra situación posible es que entre la intersección de las zonas se encuentren tres o más ejes distintos uno después de otro. Con la solución expuesta, solo se detectan dos pero es un error que se considera dentro de los parámetros aceptables.

En la Figura 5 se puede apreciar: en verde las manzanas calculadas, en gris las zonas censales, en azul los ejes de calles y las perpendiculares trazadas a la intersección de las zonas que interceptan los ejes para su detección.

Una vez que obtengo los ejes que median entre dos zonas, lo debo de repetir para cada zona lindera de la zona tomada inicialmente.

Hasta aquí para automatizar esta tarea se crea un script que obtiene todas las zonas de la capa de zonas y en loop toma una a una. En el cuerpo de la iteración se vuelve a obtener



Figura 5 - zonas ejes

todas las zonas de la capa excepto la actual zona en la que va la iteración primera y también entra en loop sobre estas zonas. Dentro del cuerpo del segundo loop se considera zona lindera si hay intersección entre la actual de loop primero y la actual del segundo loop. Si hay intersección se acciona con la solución detallada.

En este punto, si tomamos una zona censal, ya sabemos que ejes la circunvalan y por lo tanto que ejes circunvalan una manzana. Resta llevar los puntos correctamente a la ubicación que corresponde en el perímetro de la manzana en función de cada eje calle. Se puede observar que básicamente se necesita ubicar en el perímetro de la manzana calculada la proyección de la geometría intersección entre las zonas censales linderas a la zona inicial. Para esto se reutiliza el cálculo de la línea intersección entre zonas censales linderas.

Se sabe que las distancia<sup>13</sup> entre el perímetro de la manzana y el perímetro de la zona censal son 6 metros, porque así se construyó. Por lo que si se calcula, un buffer (elemento amarillo en la Figura 6) mayor a 6 metros a la geometría intersección entre zonas censales linderas, el resultado de la intersección del buffer con la manzana será un tramo del perímetro de esta (línea roja en la Figura 6), que es lo que queremos hallar. Este, es el “tramo proyección”<sup>14</sup> que se corresponde con la geometría intersección de las zonas linderas y que además ya sabemos a qué calle se corresponde.

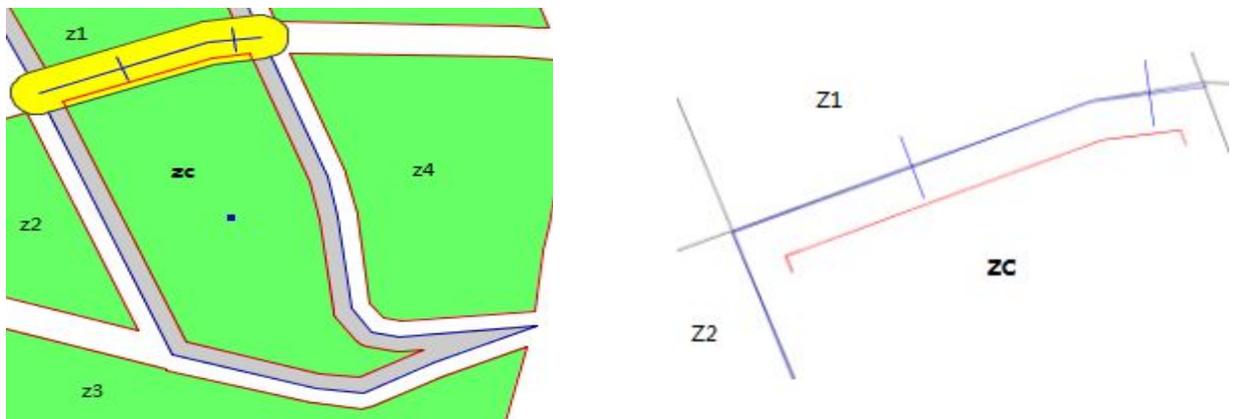


Figura 6 - tramo en manzana

Haciendo uso de las estructuras ordenadas de puntos, se puede saber cuántos y cuáles puntos se corresponden con el tramo del perímetro de la manzana que se halló, ya que entre los últimos cálculos obtuvimos a qué calle/s corresponde dicho tramo.

Es imposible calcular la ubicación precisa de los puntos de direcciones, es decir, la puerta de calle del domicilio, pero para cumplir el objetivo vamos a modelar la realidad a una solución que se aproxima.

Se considera como un error tolerable distribuir los puntos de direcciones que comparten número de puerta de forma equidistante a la que pertenecen en la cuadra. Se puede notar, que en esta definición de tolerancia de error, se hace referencia al concepto cuadra. Con anterioridad se usó el término tramo para referirse al mismo concepto, concepto que en la *sección 2 El problema* no se pudo definir y ahora en base a cálculos se puede trabajar con él.

Para ubicar los puntos de direcciones en su cuadra se toma el tramo del perímetro de la

<sup>13</sup> Abusando de la definición de distancia.

<sup>14</sup> No se desea dar formalidad sobre la expresión, sino manejar la idea intuitiva de la misma.

manzana obtenido y se calcula la distancia entre los extremos, sin importar qué forma tenga. Como sabemos que calle/s corresponde al tramo, a partir de las estructuras de puntos podemos calcular la cantidad de los mismos. Realizando el cociente entre la distancia calculada y la cantidad de puntos que no comparten número de puerta + 1, se obtiene que distancia debe haber entre puerta de calle con distinto número. Con este valor se hacen tantos buffers (en rojo en la Figura 7) como cantidad de puertas de calle en tramo. Para ello se parte del punto inicial del tramo y se obtiene el corte entre el buffer y dicho tramo (de izquierda a derecha en la Figura 7). Se toma el último corte como el centro del próximo buffer, luego se repite el procedimiento anterior tantas veces como puertas de calle pertenecen al tramo. A partir de las estructuras se puede ubicar cada punto o conjunto de puntos (en caso de ser edificio) en cada punto de corte actualizando su geometría. Luego se repite los últimos pasos con el resto de las zonas censales linderas de la zona tomada inicialmente resolviendo finalmente el problema inicial.

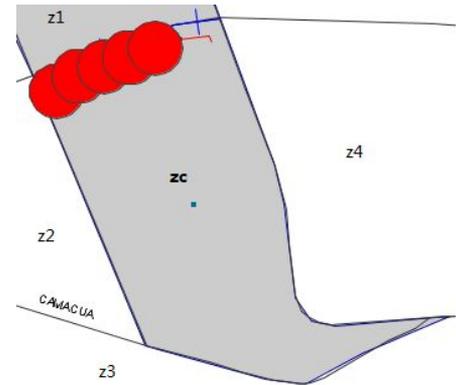


Figura 7 - ubicación de puntos

Hasta el paso anterior las operaciones a nivel de script son transparentes y literales de la explicación. La dificultad reside en calcular los cortes del tramo hallado. Para eso se hace un loop de 1 hasta la cantidad de domicilios que corresponda, y ubicado en el punto inicial del tramo se realizan buffers -de la distancia calculada-. En la primera iteración solo habrá un punto de corte, ese será el próximo centro del buffer de la próxima iteración, de aquí en mas se continúa con el loop verificando que el buffer tendrá dos puntos de corte con el tramo, el buscado y el centro del buffer anterior, el cual se descarta. En cada corte, a partir de la estructura de puntos y a través de los identificadores se obtiene la, o las entidades punto de la capa c\_puntos que corresponden con el corte y se actualiza la geometría de estos con la geometría resultante del corte del buffer y el tramo.

A continuación, en la Figura 8 se muestra el resultado de una ejecución exitosa. Se puede observar en rojo los puntos con la geometría original: ubicados en el centroide de su zona censal. En negro los puntos anteriores reubicados, producto de la ejecución de la solución. En verde claro las manzanas calculadas. En verde oscuro las zonas censales y en azul los ejes de calles.



Figura 8 - resultado final

## 6. Errores: tolerancia, identificación y mitigación.

Con la solución adoptada se introducen errores que se deben analizar y establecer si están dentro de los parámetros tolerables, en este apartado se identifican, se definen la tolerancia y se intentan mitigar..

Un error que ya se expuso en la *sección 5 La solución*, es la ubicación de los puntos dentro de la cuadra a la que pertenecen. Se aclaró que el criterio de ubicación para los puntos agrupados por número de puerta es que sean equidistantes dentro de la geometría calculada correspondiente a su cuadra. Al respecto, por no contar con mas datos que los presentados en la *sección 3 Hipótesis iniciales*, se entiende que el criterio seguido y por tanto la solución es la mejor aproximación que se puede hallar. Por lo que el error en cuestión se considera como tolerable.

En la *sección 3*, también se nombra la cantidad de calles que puede haber entre dos zonas linderas. Esto refiere a ejes de calles que pertenecen a la capa `c_ejes_calles` y que se ubican uno a continuación del otro y que son entidades completamente distintas. En un escenario como el descrito, la solución hallada detecta hasta dos calles distintas. El universo de datos indica que puede haber incluso mas de tres.

Poder encontrar mas de dos calles plantea al menos dos problemas:

1. cómo hacerlo, y definir si se quiere detectar todas la calles o algún número determinado.
2. el ordenamiento espacial de qué calle va primero a cual (una vez resuelto el punto 1.)

Para el problema 1. se pueden encontrar soluciones muy complejas que pueden pasar por dividir la geometría intersección de las zonas linderas en muchas partes en función si se encuentran calles nuevas o no, e ir trazando perpendiculares para hallar los ejes, siguiendo un esquema semejante al planteado en este artículo.

El problema 2. agrega complejidad de ordenamiento entre calles, ya que estas definen la ubicación de los puntos. No ordenar correctamente las calles es igual a no ordenar correctamente los puntos.

La solución se puede lograr usando algoritmos de ordenamiento típicos aunque con la complejidad de ordenar mas de dos elementos (calles).

Se encuentra tolerable adoptar una solución como la presentada que detecte hasta dos calles en situaciones como la descrita, ya que se entiende que es mucho el esfuerzo para la corrección en comparación con la mejora agregada.

Otro error encontrado por el enfoque adoptado es: primero, una zona censal (zc en la Figura 9) tiene una zona linderas (z2 en la Figura 9) que define una calle en su intersección con la primera. El algoritmo construido, tomando en cuenta dicha calle y z2 ubica los puntos donde corresponde (línea roja en la Figura 9). Luego, si una segunda zona linderas (z1 en la Figura 9), define una calle que coincide con la calle hallada inicialmente se experimenta un comportamiento no del todo deseable. El script desarrollado, toma los puntos que se ubicaron en primera instancia, y los va a reubicar en función de los cálculos de la segunda zona vecina linderas (línea azul en la Figura 9). El comportamiento ideal sería redistribuir los puntos en la unión de las geometrías de las cuadras halladas ( la unión de la línea roja y azul en la Figura 9). El error podría subsanarse manteniendo una estructura alternativa que guarde la geometría hallada a partir de las zonas linderas y las unifique cuando estas tengan una calle en común. Se debe manejar una estructura ya que no se sabe en qué orden se itera sobre las zonas linderas a la tomada inicialmente por lo que no sería suficiente una variable.

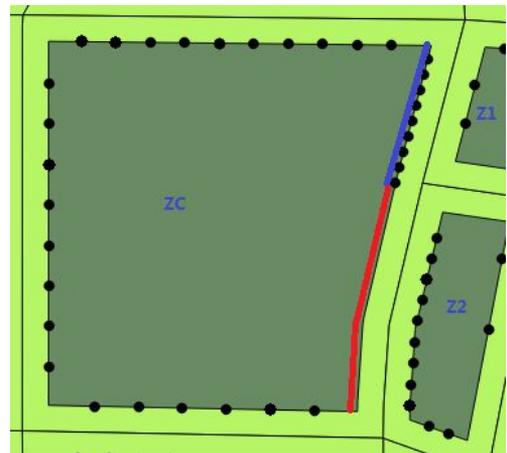


Figura 9 - error zonas linderas

Nuevamente se entiende tolerable este error en función del esfuerzo de la solución y la mejora agregada.

Un error que incide de forma transversal a este proyecto es el error en los datos, o como habitualmente se lo llama: la calidad de datos. En general la mayoría de los datos son construidos por seres humanos, por lo tanto son permeables a errores de distinta índole como ser: faltas de ortografía, abreviaciones incorrectas, o simplemente no seguir un estándar. Al momento de hacer corresponder el nombre de una calle con la calle de un punto, a nivel de script, lo único que hay detrás es una comparación entre cadenas de caracteres. Por lo que una falta ortográfica insignificante en el nombre de la calle de un punto, puede redundar en no encontrar una calle con el mismo nombre (mal escrito). Por ejemplo, si un punto pertenece a la calle mal escrita “VRECHA” nunca se lo podrá asociar a la calle con nombre correcto “BRECHA”.

Para mitigar este tipo de errores se implementó un diccionario-corrector basado en probabilidades y se introdujo en la solución general. El diccionario tiene dos argumentos, una fuente de palabras, en este caso el nombre de las calles que tienen asociados los puntos de una manzana, y el segundo argumento es una palabra a corregir (basado en la fuente de palabras).

El algoritmo solución para determinar qué puntos de una zona censal se asocian a una cuadra, compara el campo nombre\_cal de los puntos, con el atributo nombre\_11 de las calles halladas como circunvalantes a dicha zona. En caso de no haber coincidencia se pasa el nombre de la calle al diccionario implementado como palabra a corregir y se toman como fuente de palabras el nombre de las calles de los puntos de la zona. El diccionario-corrector tiene un alto grado de acierto.

Asociado al contexto anterior, para mitigar los errores de falta de abreviaciones o sufijos, como: “AVENIDA”, “GENERAL”, “GRAL”, etc, se implementó dentro de la solución general una función que si no encuentra una asociación entre la calle de un punto y el nombre de una calle, combina sufijos como: “AV”, “AVENIDA”, “GRAL”, “GENERAL”, etc, hasta encontrar una eventual coincidencia.

La solución anterior mejora los resultados, pero degrada mucho la performance de la ejecución. Por lo que se opta por deshabilitar este componente del script.

Otros errores que se encontraron que forman parte de la calidad de datos y que no fueron resueltos o mitigados son: errores de geometría en las capas, falta de ejes de calles entre zonas censales linderas y no correspondencia entre el nombre de calle de los puntos y el nombre de la calle a la cual pertenecen.

Un aspecto que puede conducir a errores es identificar qué entidades de la capa de puntos han sufrido cambios, esto es: determinar los puntos que fueron afectados por el procedimiento y por tanto tienen una nueva geometría asociada de los que no. Si no se resuelve este ítem, no se puede advertir el porcentaje de éxito de una ejecución. Incluso, dicha información es necesaria para un eventual tratamiento sobre los puntos no afectados.

Este problema se resolvió completamente modificando el esquema de la capa de puntos. Se agregó un campo booleano inicializado en “False”. Los puntos que fueron exitosamente modificados se les seteo “True” en dicho atributo. Por lo que con un simple filtro sobre el campo se puede identificar los puntos modificados de los que no.

## 7. Debugging

En todo desarrollo de software es importante identificar y corregir errores. El debugging en el mundo SIG amplifica su utilidad si se incorpora el efecto espacio-visual. Esto es, en el momento que un script realiza un cálculo auxiliar como un buffer, una línea o algo con geometría, se puede agregar como gráfico o entidad virtual al espacio donde se visualizan las capas. De este modo se puede visualizar dichas cálculos, detectar errores o simplemente contribuir al desarrollo de una mejor solución con la nueva perspectiva.

En la Figura 10 se puede observar una ejecución con los componentes desarrollados de debugging activos.

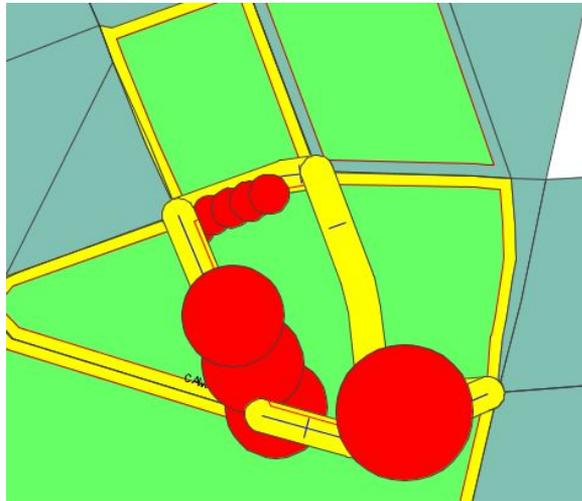


Figura 10 - debugging

## 8. Conclusión

Luego de un estudio basado en un caso propuesto se establece y caracteriza un problema propio de los sistemas de información geográfica (SIG).

El aporte de los SIG es indiscutible frente a los sistemas de información tradicionales (SI). Los SIG ofrecen (informalmente) características espacio-visuales por sobre las tecnologías tradicionales de SI. Este artículo enmarca la dificultad que aparece al querer emplear dichas características para resolver eventuales problemas dentro de un escenario SIG como ser el presentado. Además se establece la causa: relaciones con componentes espacio-visuales difíciles de caracterizar y por tanto mapear, por ejemplo a un script (si se quiere automatizar una solución).

También se caracterizó una solución a este tipo de problemas: servirse de propiedades espaciales entre los datos y las propias entidades espaciales para encontrar cálculos intermedios que determinen la solución buscada, por lo que puede resultar en problemas no resolubles, la solución también está dada por el ingenio del involucrado. Se entiende que este último punto da valor a este artículo, ya que el acopio de soluciones particulares como las que se describieron en la *sección 5 La solución*, sirven de base para ampliar el horizonte para otros eventuales problemas de igual característica.

Sobre la definición del margen de error sobre una solución: concluimos que es un problema en sí pero inverso al que se viene tratando, ya que se parte de un valor georeferenciado esperado, un valor georeferenciado obtenido (solución), luego se debe encontrar una propiedad que los relacione, para finalmente, encontrar un parámetro que caracterice el error tolerado, en caso de que se desee proceder formalmente. Un ejemplo puede ser que la solución calculada esté dentro de un buffer de un radio a determinar de la solución ideal.

Se cumplió el objetivo de construir una herramienta que resuelve el problema general planteado en la *sección 4 El problema*, y que permite conocer el potencial que tiene el módulo de scripting de gvSIG 2.2 Desktop. Algunas conclusiones al respecto sobre dicha herramienta:

- es una buena elección como herramienta por lo sencillo de su aprendizaje.
- tiene un editor con apenas las funciones básicas.
- es inestable.

- la performance es muy baja, la edición de capas es muy lenta.
- la comunidad activa que da soporte permitió evacuar dudas con rapidez.

De los ítems anteriores, se sigue primando el primero lo que da un resultado positivo.

Finalmente, con todos los puntos expuestos en esta sección se concluye que se alcanzaron los objetivos planteados en la *sección 1.2 Objetivos*.

## 9. Referencias bibliográficas

- Correo Uruguayo, <http://www.correo.com.uy>.
- Documentación de QGIS2.8, [http://docs.qgis.org/2.8/es/docs/pyqgis\\_developer\\_cookbook](http://docs.qgis.org/2.8/es/docs/pyqgis_developer_cookbook), fecha consulta: marzo 2016.
- gvSIG asociación, <http://devel.gvsig.org/sites>, fecha consulta: abril 2015.
- gvSIG blog, <https://blog.gvsig.org>, fecha consulta: marzo 2016, fecha actualización sitio web: 1 noviembre 2016.
- INE - Unidades Geoestadísticas - Uruguay, <http://www.ine.gub.uy/documents/10181/18006/definiciones+para+web.pdf/896410b7-f7c2-40f0-b5c3-4d7b7326f51c>, fecha consulta: marzo 2015.
- Mappinggis, <http://mappinggis.com>, fecha consulta: marzo 2016.
- másquesig, <https://masquesig.com>, fecha consulta: marzo 2016, fecha actualización sitio web: 3 noviembre 2016.