Utilización de *software* de Big Data en la arquitectura y la planificación urbano-territorial

Dante Andrés Barbero
Posdoctorado (UNC). Doctor en Ciencias Informáticas (UNLP). Magíster en Paisaje, Medio Ambiente y Ciudad (UNLP). Licenciado en Informática (UNLP). Analista de computación (UNLP). Investigador adjunto del CONICET. IIPAC, CONICET-UNLP.
dantebarbero@yahoo.com.ar

Pedro Joaquín Chévez
Doctor en Ciencias (área: Energías Renovables) (UNSa). Especialista en Energías Renovables (UNSa). Arquitecto (UNLP). Becario posdoctoral del CONICET. Docente FAU, UNLP. IIPAC, CONICET-UNLP.
che.pedro@hotmail.com

Carlos Alberto Discoli
Doctor en Ciencias (área: Energías Renovables) (UNSa). Magíster en Ambiente y Patología Ambiental (UNLP/Universidad de Siena). Especialista en Ambiente y Patología Ambiental (UNLP/Universidad de Siena). Ingeniero Mecánico (UNLP). Investigador principal del CONICET. IIPAC, CONICET-UNLP.
discoli@rocketmail.com

Irene Martini
Doctora en Ciencias (área: Energías Renovables) (UNSa). Magíster en Políticas Ambientales y Territorales (UBA). Arquitecta (UNLP). Investigadora independiente del CONICET. Docente FAU, UNLP. IIPAC, CONICET-UNLP.
irene_martini@yahoo.com.ar

http://dx.doi.org/10.30972/crn.29294624
ISNN 1666-6186. Volumen 29 N.° 29 (DIciembre de 2020) Pp. 099-118 - Recibido: 15-04-20. Evaluado y aprobado: 06-09-20

Utilización de *software* de Big Data en la arquitectura y la planificación urbano-territorial

Resumen

Un problema recurrente en arquitectura y planificación urbano-territorial es poder encontrar grupos de elementos con características homogéneas. En arquitectura, las clasificaciones edilicias/constructivas se deducen a partir de un número de parámetros o variables, y si se analiza la estructura urbana es posible identificar áreas homogéneas según el tipo de uso de suelo, cobertura de servicios, entre otros aspectos. Cuando el volumen de datos para procesar es tal que no se pueden analizar mediante métodos convencionales, es necesario recurrir a técnicas de Big Data. En este trabajo se utilizará un *framework* para Big Data (Apache Spark) para descubrir áreas homogéneas en cuanto a cobertura de servicios urbanos básicos de infraestructura y saneamiento. Identificar tales áreas permitirá localizar lugares con similares prestaciones, inferir nuevas demandas en función de posibles crecimientos urbanos e identificar lugares de la periferia hacia donde puede crecer la ciudad, entre otros posibles usos.

Palabras clave

Big Data; minería de datos; arquitectura; planificación urbana.

Use of Big Data software in architecture and urban-territorial planning

Abstract

A frequent problem in architecture and urban-territorial planning is to be able to find groups of elements with homogeneous characteristics. In architecture, building/construction classifications are deduced from a number of parameters or variables; and if the urban structure is analyzed, it is possible to identify homogeneous areas according to the type of land use, services coverage, among other possible aspects. When the volume of data to be processed is such that it cannot be analyzed by conventional methods, it is necessary to use Big data techniques. In this work, a framework for Big data (Apache Spark) will be used to discover homogeneous areas in terms of coverage of urban basic services of infrastructure and sanitation. Identifying such areas will allow to locate places with similar benefits, infer new demands based on possible urban growths and identify places on the periphery where the city can grow, among other possible uses.

Keywords

Big Data; Data Mining; architecture; urban planning.

Uso de *software* de Big Data em arquitetura e planejamento urbano-territorial

Resumo

Um problema recorrente na arquitetura e no planejamento urbano-territorial é conseguir encontrar grupos de elementos com características homogêneas. Na arquitetura, as classificações edilícias / construtivas são deduzidas de uma série de parâmetros ou variáveis, e se a estrutura urbana for analisada é possível identificar áreas homogêneas de acordo com o tipo de uso do solo, cobertura de serviços, entre outros aspectos. Quando o volume de dados a processar é tal que não pode ser analisado pelos métodos convencionais, é necessário recorrer às técnicas de Big Data. Neste trabalho, será usado um framework para Big Data (Apache Spark) para descobrir áreas homogêneas em termos de cobertura de infraestrutura urbana básica e serviços de saneamento. Identificar tais áreas permitirá localizar lugares com benefícios semelhantes, inferir novas demandas a partir do possível crescimento urbano e identificar lugares na periferia onde a cidade pode crescer, entre outros usos possíveis.

Palavras chave

Big Data; mineração de dados; arquitetura; planejamento urbano.

Introducción

Big Data es la combinación de varias tecnologías orientadas a facilitar el tratamiento de grandes volúmenes de datos. En la actualidad, algunas aplicaciones requieren almacenar grandes volúmenes de datos que en ocasiones, debido precisamente a su tamaño, no se pueden procesar, almacenar y analizar mediante métodos convencionales. En este punto, una alternativa posible es recortar el universo de análisis (perdiendo algunas variables o incluso dimensiones enteras); la otra alternativa es usar Big Data.

En la actualidad, existen trabajos en los que se ha utilizado Big Data para resolver problemas relacionados con la sostenibilidad urbana (Wang et al., 2018); la planificación urbana (Thakuriah et al., 2017) y regional (Schintler et al., 2018). Otros trabajos (Zhou et al., 2016; Koseleva et al., 2017) han utilizado Big Data para el análisis de servicios urbanos básicos. Estos dos últimos trabajos presentan un *framework* para analizar el comportamiento del consumo energético en las viviendas desde tres dimensiones: usuario, tiempo y espacio.

En la arquitectura y la planificación urbano-territorial aparece frecuentemente el problema de encontrar grupos de elementos con características homogéneas. Por ejemplo, las tipologías edilicias, las clasificaciones de viviendas desde el punto de vista constructivo, los catálogos de paisajes, la morfología urbana, los tipos de uso de suelo, entre otros, se deducen a partir de un número de parámetros o variables. Cuando el número de elementos (datos) con los que se trabaja es tal que no se pueden analizar mediante métodos convencionales, es necesario recurrir a técnicas de Big Data. En este trabajo se realizará un análisis para identificar grupos de objetos con características homogéneas en cuanto a cobertura de servicios urbanos básicos de infraestructura y saneamiento, utilizando para ello un *framework* para Big Data denominado Apache Spark. Para tal fin se desarrolló un *script* en lenguaje Python que implementa el método de agrupamiento/clústering K-means. Poder identificar áreas homogéneas en cuanto a la cobertura de servicios permite localizar en el territorio lugares con prestaciones similares y a partir de ello inferir, por ejemplo, nuevas demandas en función de posibles crecimientos urbanos, lugares de la periferia hacia donde puede crecer la ciudad, entre otros usos y aplicaciones.

Para la ejecución de aplicaciones de Big Data, es necesario contar con *hardware* (ejemplos: clústers, sistemas distribuidos, *cloud computing*) y *software* específicos.

Las soluciones a problemas de Big Data suelen lidiar con las cuatro V: volumen, velocidad, variedad y valor. Estas cuatro características hacen referencia al **volumen** de información (por lo general, capacidades de almacenamiento que exceden las de un solo equipo y requieren el almacenamiento en varios equipos); la **velocidad** de procesamiento, que puede aumentarse si los datos se dividen y procesan en paralelo o de forma distribuida; la **variedad** en el sentido de que los programas de Big Data deben poder trabajar con distintos tipos de datos (ejemplos: no estructurados, semiestructurados y estructurados) y por último el **valor**, que hace referencia a poder sacar provecho de grandes volúmenes de datos (por ejemplo: extrayendo conocimiento a partir de estos). Precisamente, extraer conocimiento a partir de datos es el objetivo de una rama de la informática conocida como minería de datos. Se entiende por minería de datos al proceso de extraer conocimiento útil y comprensible, previamente desconocido, desde grandes cantidades de datos almacenados en distintos formatos (Witten et al., 2000).

> Este conocimiento puede ser en forma de relaciones, patrones o reglas inferidos de los datos y (previamente) desconocidos, o bien en forma de una descripción más concisa (es decir, un resumen de los mismos). Estas relaciones o resúmenes constituyen el modelo de los datos analizados. Existen muchas formas diferentes de representar los modelos y cada una de ellas determina el tipo de técnica que puede usarse para inferirlos (Hernández Orallo et al., 2004, p. 12).

Los modelos pueden ser de dos tipos: **predictivos** o **descriptivos**. Los primeros intentan estimar valores futuros o desconocidos de variables de interés (**variables objetivo** o **dependientes***)*, usando para ello otras variables o campos de la base de datos (las **variables independientes** o **predictivas***)*. Los modelos descriptivos sirven para identificar patrones que explican o resumen los datos, es decir, realizan una tarea exploratoria sobre los datos, en lugar de una predictiva (Hernández Orallo et al., 2004). En minería de datos, son ejemplos de modelos predictivos la clasificación y la regresión, en tanto que ejemplos de los modelos descriptivos serían las reglas de asociación, el agrupamiento/clústering y el análisis correlacional. El modelo utilizado en este trabajo es de tipo **descriptivo** y utiliza el algoritmo de **agrupamiento/clústering K-means**.

En cuanto a los datos con los que se trabaja en Big Data, existe *software* para tratar con datos estructurados (como podrían ser las bases de datos relacionales); semiestructurados (ej: archivos de texto y hojas de cálculo) y no estructurados (ej.: imágenes, fotos, audio y video). Estos datos, a su vez, pueden ser procesados en forma *batch* (se guarda todo y luego se procesa) o *streaming* (*on line*, en tiempo real o cercano a tiempo real). En este trabajo se

utilizó el *framework* Hadoop, que soporta procesamiento de grandes bases de datos en un ambiente distribuido. Incluye un sistema de archivos distribuidos (HDFS, Hadoop distributed file system). Hadoop permite procesamiento *off line* (*batch*). En dicho *framework*, los diferentes nodos que componen un clúster "ven" un único sistema de archivos compuesto por las distintas unidades de almacenamiento de cada uno de los nodos como si fuese uno solo (y cada nodo ve lo mismo), perdiendo un poco la noción de la ubicación física en la cual se encuentra realmente almacenado un dato. Todo el contenido del clúster se ve como un único sistema de archivos (*file system*): el HDFS. Sobre Hadoop es posible ejecutar aplicaciones de alto nivel, como Cascading, Pig, Hive y Spark, entre otras.

Spark es un *framework* de procesamiento de datos de propósito general (Apache Software Foundation, 2019a), y soporta los dos modos de trabajo en Big Data: procesamiento *batch* (en Apache Spark) y procesamiento de flujos de datos (o *streaming* de datos) (en Apache Spark Streaming). Spark fue desarrollado para que trabaje con el HDFS de Hadoop, pero también permite la integración con otros medios de almacenamiento distribuido, tales como Hbase, Google Cloud, Cassandra, MongoDB y Amazon S3, entre otros. Es posible desarrollar aplicaciones para Spark en diferentes lenguajes de programación, como Java, Python, R y Scala. Además, posee una gran cantidad de APIs (sigla de *Application Programming Interface*) desarrolladas para trabajar con grandes volúmenes de datos, y está optimizado para trabajar en memoria RAM. Spark incluye una librería de algoritmos de *machine learning* llamada MLlib (librería de *machine learning*). MLlib es una librería que viene incluida en Apache Spark que contiene numerosos algoritmos para resolver problemas de clasificación, regresión, árboles de decisión, sistemas recomendadores, *clustering*, entre otros (Apache Spark Foundation, 2019b). Estos algoritmos pueden ejecutarse en modo distribuido en un clúster de PC o localmente en una sola PC o similar que cuente con múltiples núcleos (*multicore*). El algoritmo para ejecutar localmente o de manera distribuida no cambia. Es decir, se debe pensar cómo resolver un problema sin tener en cuenta dónde se va a ejecutar. Así, aspectos como la distribución de los datos en los diferentes nodos, la sincronización entre procesos y otras cuestiones que hacen a la ejecución de los algoritmos corren por cuenta del *framework* Spark delegando en el programador la tarea de encontrar una solución al problema que pretende resolver. De esta manera, dentro de la librería MLlib, incluida en Apache Spark, se pueden encontrar algoritmos de agrupamientos y, entre ellos, el utilizado en este trabajo: K-means (Wu et al., 2008).

Una aplicación Spark está compuesta por un módulo principal denominado **driver**, que es el encargado de "paralelizar" los datos (dividir la aplicación del usuario en

varias operaciones en paralelo sobre un clúster de computadoras) y enviar los comandos correspondientes a cada nodo del clúster. Spark también permite el uso de variables compartidas (cada nodo del clúster tiene una copia de tales variables). Gracias a esta característica es posible implementar el método K-means (Mac Queen, 1967), de modo que, al distribuir el conjunto de datos (que agrupar) entre los diferentes nodos para analizar cuál es el centroide[1] más cercano, la lista de coordenadas de cada centroide sea la misma en todos los nodos del clúster. La distancia de cada objeto al centroide se calcula utilizando la fórmula de la distancia euclídea n-dimensional.

1. Se entiende por centroide el vector cuyas coordenadas representan el centro geométrico de un objeto n-dimensional.

Spark también soporta un modo local pseudo-distribuido, normalmente utilizado solamente para pruebas o en entornos de desarrollo donde el almacenamiento distribuido no es obligatorio y se puede usar el sistema de archivos local (en una única computadora). En un escenario así, Spark se ejecuta en una única máquina dividiendo la tarea en los diferentes núcleos (*cores*) de un procesador de múltiples núcleos (*multi-core*).

En este trabajo se plantea crear una aplicación escrita en lenguaje Python, que utilizará la librería MLlib y se ejecutará en el *software* Spark, el cual, a su vez, trabajará con el sistema de archivos de Hadoop.

Metodología

En ocasiones, un elemento u objeto de estudio suele ser descripto por medio de una serie de atributos. Cuando se tiene una gran cantidad de elementos y, a su vez, cada uno es descripto por un número significativo de atributos, encontrar grupos de elementos que presenten características similares no es una tarea sencilla. En este tipo de problemas es necesario aplicar metodologías que permitan extraer conocimiento a partir de grandes volúmenes de datos, y la minería de datos es la rama de la informática que se ocupa de ello. Como se mencionó, dentro de la minería de datos existen técnicas predictivas y explicativas. Dentro de las técnicas explicativas se encuentran los **métodos de agrupamiento** (*clustering*), que permiten reconocer clases de objetos donde los elementos que pertenecen a cada una presentan características similares, y de allí su importancia en diferentes disciplinas. En particular, en la arquitectura y la planificación urbana, los métodos de agrupamiento pueden utilizarse para detectar viviendas con tipologías similares o con características socioeconómicas parecidas, entre otros usos. En estudios que incluyen aspectos relacionados con el territorio, pueden aplicarse, por ejemplo, para encontrar usos de suelo de similares características, detectar vacancias, etc. Por otra parte, cuando el

número de filas (registros) es superior a 1.048.576, las planillas de cálculo como Microsoft Excel o LibreOffice Calc no admiten la posibilidad de trabajar con tal cantidad de información. Por este motivo, y para resolver el problema de almacenamiento de información, se utilizó el HDFS de Hadoop. Para poder aplicar algoritmos de manera rápida a los datos almacenados en el clúster, se utilizó Apache Spark. Más específicamente, se utilizó el método de agrupamiento K-means que se encuentra en la biblioteca MLlib y que se instala automáticamente junto con Apache Spark. Para ello se creó un *script* en Python que usa el método K-means incluido en la librería MLlib y se ejecuta sobre el *software* Apache Spark. Spark, a su vez, lee los datos que debe procesar desde el sistema de archivos Hadoop.

Como ejemplo de aplicación se analizaron los datos de cobertura asociados a los servicios básicos de agua, cloacas, gas y electricidad en cada uno de los radios censales[2] de la ciudad de La Plata (INDEC, 2010). Por cada radio censal se calcularon previamente los porcentajes de viviendas (respecto del total de viviendas del mismo radio censal) que contaban con los servicios de agua, gas, electricidad y cloaca por red. Por lo tanto, los datos seleccionados para ejecutar el algoritmo de *clustering* tienen la forma de un vector de cuatro elementos, los cuales se mencionan a continuación:

(% de viviendas con saneamiento cloacal por red , % de viviendas con gas natural por red,
% de viviendas con agua por red, % de viviendas con electricidad por red)

De esta manera, el análisis permite identificar, agrupar y localizar con precisión las áreas urbanas con similares características, visibilizando vacancias en los porcentuales de cada cobertura y sus diferentes combinaciones. Dado que cada radio censal es representado como un vector 4-dimensional y se asociará al centroide más cercano, es decir, a aquel centroide que aplicando la fórmula de la distancia euclídea n-dimensional esté a menor distancia, es de esperar que los valores de cada coordenada de un radio censal estén "cerca" del valor que tiene, en esa misma coordenada, el centroide más cercano.

Cabe destacar que, si bien en este caso el número de registros fue inferior al millón, la solución de Big Data utilizada es la misma y funcionaría correctamente con millones de registros, es decir, la solución es escalable. De modo que si, por ejemplo, se quisieran analizar características similares considerando todos los radios censales del país, o incluir otras variables o dimensiones, la solución aquí presentada no solo sería la apropiada, sino que además sería la única forma de poder tratar semejante volumen de datos.

2. Un radio censal es un área correspondiente a una unidad geográfica empleada en el censo y que comprende aproximadamente 300 viviendas en las ciudades, en tanto que en zonas rurales el número de viviendas considerado es menor. En el presente trabajo se utilizaron los radios censales del censo 2010.

Resultados

Los resultados obtenidos al aplicar K-Means para encontrar grupos desde dos hasta diez clases con características homogéneas en cuanto a prestación de servicios básicos urbanos dieron como resultado los gráficos de las figuras 1 a 5.

Como se puede observar en la figura 1, los radios censales de una misma clase tienen el mismo color. Por lo tanto, el mapa de la figura 1 muestra el resultado obtenido al considerar dos clases distintas, mientras que el mapa de la figura 1.1. muestra lo que acontece al agregar una tercera clase.

En la figura 1 se pueden apreciar las zonas urbanizadas con acceso a servicios básicos en color azul. Asimismo, si se considera la tabla 1, se puede observar que todas las coordenadas de la clase 2 presentan un alto porcentaje de cobertura en cada uno de los servicios, de lo que se deduce que se trata de la parte urbana más consolidada.

Tabla 1

Coordenadas de los centroides para k = 2 clases

Fuente: elaboración propia

% viviendas con saneamiento cloacal por red	% viviendas con gas por red	% viviendas con agua potable por red	% viviendas con electricidad por red	Número de clase	Color
11,262136	38,009709	69,529126	93,514563	1	
95,310178	92,760905	98,66559	98,864297	2	

En la figura 1.1., al introducirse un tercer grupo, se observa que los radios censales color violeta se corresponden, en general, a la periferia urbana, en tanto que los de color verde señalan zonas predominantemente rurales. Los radios censales de color marrón se corresponden, una vez más, con la parte urbana consolidada, y así lo demuestran las coordenadas del centroide de la clase 1 (tabla 2), cuyos valores son los más altos y presentan un alto porcentaje de cobertura de servicios.

Tabla 2

Coordenadas de los centroides para k = 3 clases

Fuente: elaboración propia

% viviendas con saneamiento cloacal por red	% viviendas con gas por red	% viviendas con agua potable por red	% viviendas con electricidad por red	Número de clase	Color
98,378676	95,990809	99,115809	99,141544	1	
64,509434	66,443396	93,292453	96,641509	2	
5,4914286	34,211429	66,222857	93,051429	3	

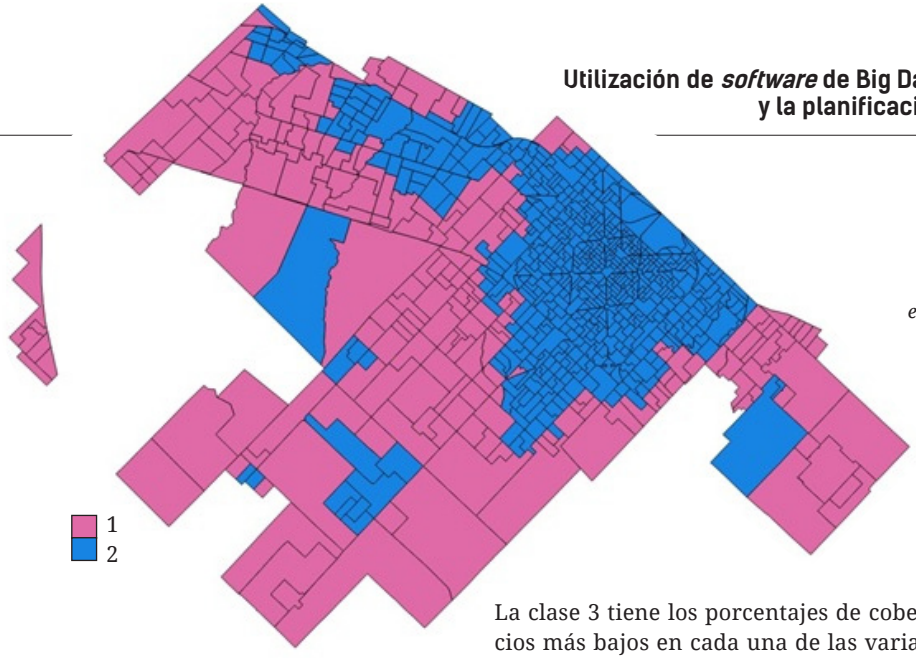

Figura 1. Resultados del clústering para k=2. Fuente: elaboración propia

La clase 3 tiene los porcentajes de cobertura de servicios más bajos en cada una de las variables, y se diferencia de la clase 2 principalmente en la baja cobertura de los tres primeros servicios.

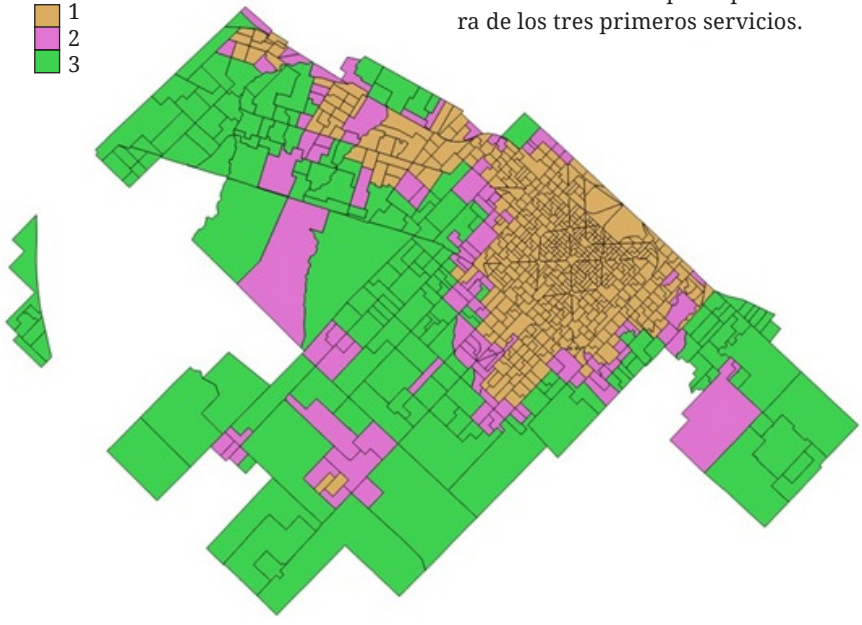

Figura 1.1. Resultados del clústering para k=3. Fuente: boración propia

Al considerar cuatro o cinco grupos (figura 2), vemos que el casco urbano se mantiene como una región homogénea de color verde para k = 4 y de color marrón claro para k = 5.

Tabla 3

Coordenadas de los centroides para k = 4 clases

% viviendas con saneamiento cloacal por red	% viviendas con gas por red	% viviendas con agua potable por red	% viviendas con electricidad por red	Número de clase	Color
7,8992248	37,674419	92,860465	93	1	
98,395948	96,029466	99,11418	99,145488	2	
67,90625	67,177083	94,395833	96,677083	3	
6,1403509	31,508772	9,4210526	93,807018	4	

Fuente: elaboración propia

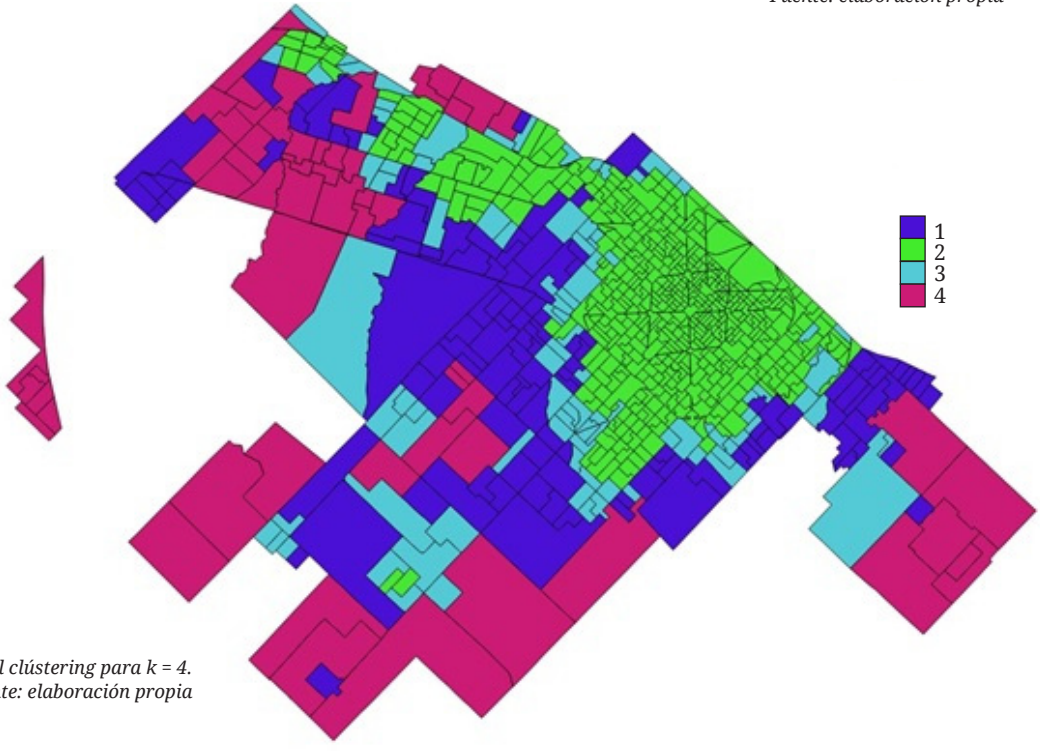

***Figura 2**. Resultados del clústering para k = 4.*
Fuente: elaboración propia

Tabla 4

Coordenadas de los centroides para k = 5 clases

% viviendas con saneamiento cloacal por red	% viviendas con gas por red	% viviendas con agua potable por red	% viviendas con electricidad por red	Número de clase	Color
6,9349593	36,203252	92,430894	92,796748	1	
3,944444	29,037037	8,3703704	93,537037	2	
53,04	79,46	87,48	98,12	3	
75,333333	54,784314	97,72549	95,333333	4	
98,276051	95,906764	99,11883	99,137112	5	

Fuente: elaboración propia

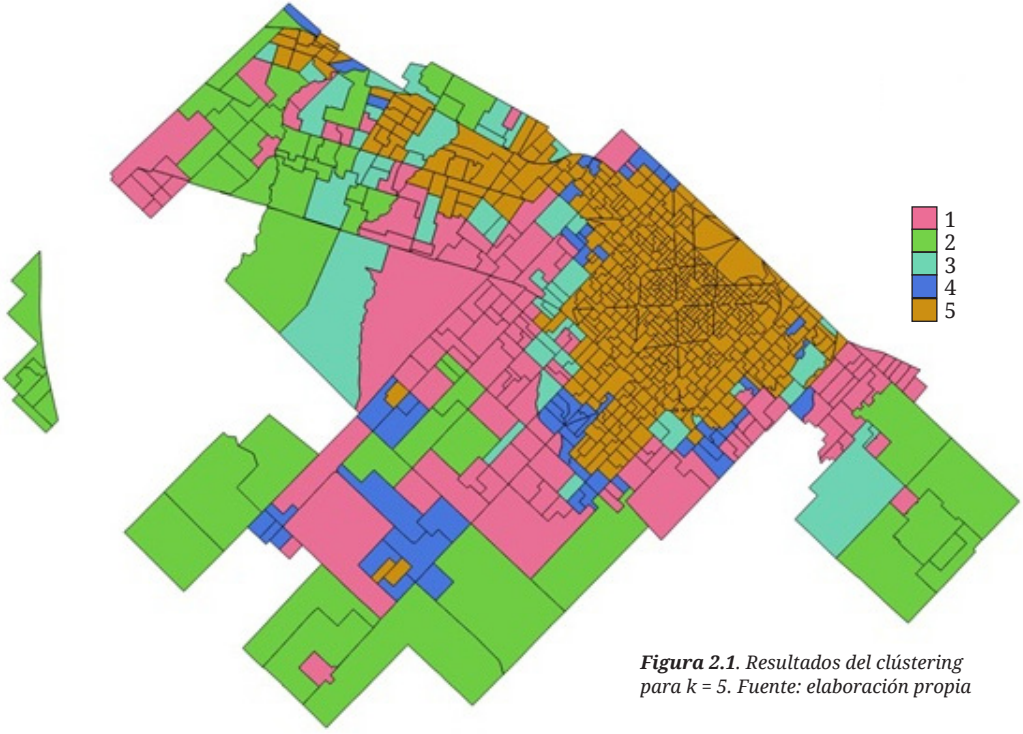

***Figura 2.1**. Resultados del clústering para k = 5. Fuente: elaboración propia*

Al considerar cuatro clases (figura 2) se observa que la clase 2 (color verde) se corresponde con la parte urbana consolidada y un alto porcentaje de cobertura en cada uno de los servicios.

Al considerar cinco clases (figura 2.1.), el casco urbano (clase 5) sigue manteniendo una alta cobertura en cada uno de los servicios y conservando homogeneidad espacial, en tanto que algunos de los sectores restantes presentan ciertos grados de fragmentación.

El resultado obtenido para k = 6 es considerado el ideal para el caso de estudio, pues se trata de un número razonable de grupos homogéneos que presentan, además, continuidad espacial en la cobertura de sus diferentes servicios. Al realizar una nueva clasificación y agregar una séptima clase, se observa que el casco urbano comienza a tener discontinuidades espaciales importantes, característica no deseable para la identificación de áreas homogéneas.

Tabla 5

Coordenadas de los centroides para k = 6 clases

% viviendas con saneamiento cloacal por red	% viviendas con gas por red	% viviendas con agua potable por red	% viviendas con electricidad por red	Número de clase	Color
61,96	38,16	97,72	93,76	1	
11,16	70,36	14,4	98,48	2	
98,426199	96,049815	99,114391	99,145756	3	
3,3428571	6,5428571	10,228571	90,8	4	
67,794872	73,820513	94,282051	97,371795	5	
6,1416667	37,416667	93,016667	92,875	6	

Fuente: elaboración propia

Tabla 6

Coordenadas de los centroides para k = 7 clases

% viviendas con saneamiento cloacal por red	% viviendas con gas por red	% viviendas con agua potable por red	% viviendas con electricidad por red	Número de clase	Color
96,024	91,12	97,56	98,816	1	
85,681818	72,015152	96,515152	96,969697	2	
51,7	64,983333	93,783333	96,583333	3	
99,432203	98,392655	99,584746	99,59322	4	
99,651163	98,534884	99,44186	97,162791	5	
5,7321429	31,053571	8,7142857	93,75	6	
6,214876	36,363636	92,46281	92,809917	7	

Fuente: elaboración propia

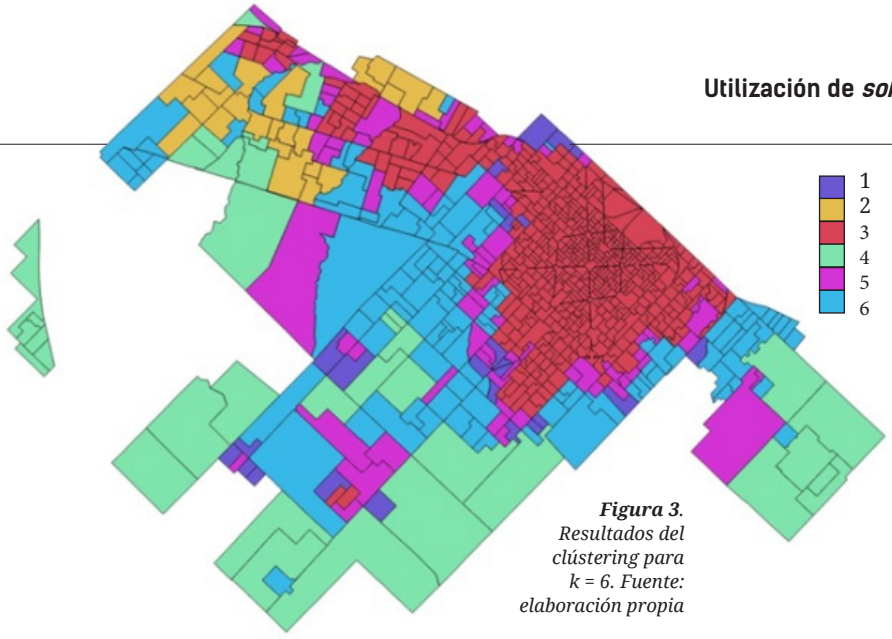

Figura 3. *Resultados del clústering para k = 6. Fuente: elaboración propia*

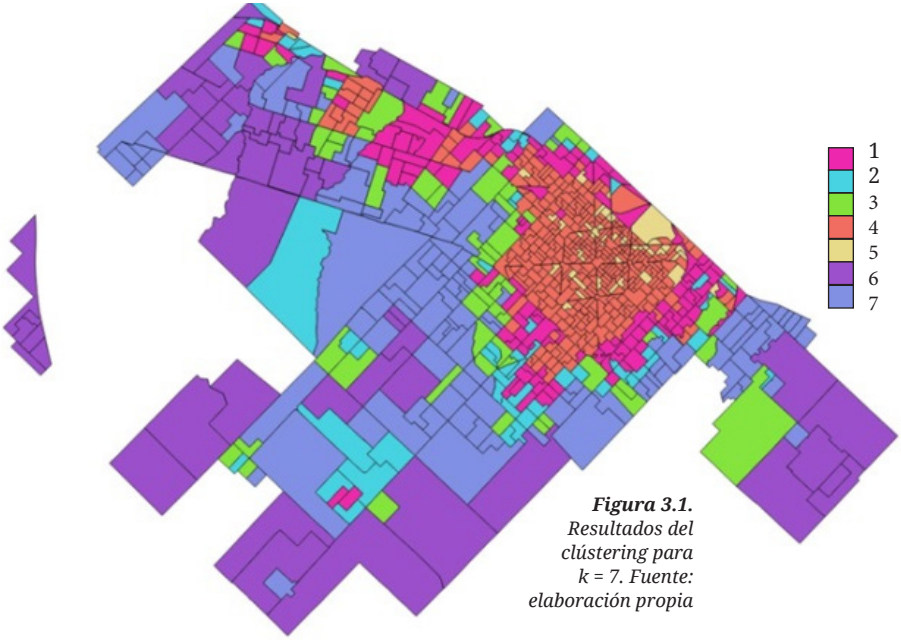

Figura 3.1. *Resultados del clústering para k = 7. Fuente: elaboración propia*

Se observa que la clase 3 de la tabla 5 se corresponde con un área que presenta continuidad espacial y alta cobertura en cada uno de los servicios. No obstante, parte de la continuidad espacial se pierde al agregar una nueva clase. Esto se ve en la tabla 6, donde se puede apreciar que las coordenadas de los centroides de las clases 4 y 5 están muy próximas entre sí y ambas clases presentan altas coberturas en cada uno de los servicios. Por lo tanto, la adición de la nueva clase (la séptima) no produjo una diferencia notoria (en cuanto a coordenadas del centroide) respecto del resto, por lo que no se justifica su existencia. En consecuencia, considerando que se desea detectar áreas con la mayor homogeneidad posible, es preferible tener seis clases en lugar de siete.

En caso de continuar el análisis incluyendo más clases (ocho o nueve clases, figuras 4 y 4.1.) ocurre que, si bien el casco urbano consolidado sigue presentando niveles altos de continuidad espacial expresando un área homogénea con un gran porcentaje de cobertura en cada uno de los servicios, estos nuevos agrupamientos aumentan la fragmentación (discontinuidad) en la periferia urbana.

En el agrupamiento con ocho o nueve clases (tablas 7 y 8) se observa que la homogeneización del casco urbano se debe a que el centroide de la clase 8 de la tabla 7 es próximo al de la clase 9 de la tabla 8, puesto que en ambos casos representan la parte urbana consolidada, de manera similar a cuando se consideraban seis clases, pero aumenta nuevamente la fragmentación (heterogeneidad) en los radios censales de la periferia.

Tabla 7

Coordenadas de los centroides para k = 8 clases

% viviendas con saneamiento cloacal por red	% viviendas con gas por red	% viviendas con agua potable por red	% viviendas con electricidad por red	Número de clase	Color
694,114583	87,416667	97,958333	98,364583	1	
57,444444	79,777778	84,388889	98,25	2	
5,7605634	19,056338	91,140845	90,464789	3	
82,785714	66,071429	97,619048	96,404762	4	
56,809524	37,714286	98,333333	93,952381	5	
3,9444444	29,037037	8,3703704	93,537037	6	
9,3965517	64,344828	94	96,396552	7	
99,277405	97,908277	99,362416	99,315436	8	

Fuente: elaboración propia

Tabla 8

Coordenadas de los centroides para k = 9 clases

% viviendas con saneamiento cloacal por red	% viviendas con gas por red	% viviendas con agua potable por red	% viviendas con electricidad por red	Número de clase	Color
10,132075	68,396226	94,509434	96,981132	1	
58,333333	80,888889	42,888889	98,555556	2	
3,9444444	29,037037	8,3703704	93,537037	3	
93,529412	93,529412	87,588235	99,352941	4	
94,541176	86,294118	99,517647	98,188235	5	
5,4078947	21,184211	91,486842	90,671053	6	
55,086957	36,695652	98,391304	93,73913	7	
73,636364	70,787879	96,712121	97,030303	8	
99,271493	97,911765	99,554299	99,31448	9	

Fuente: elaboración propia

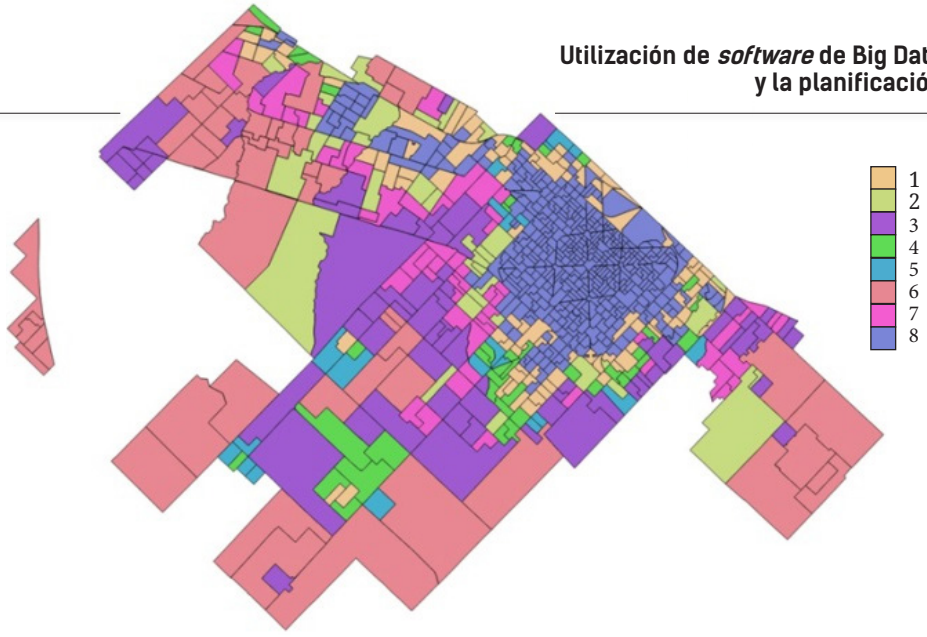

Figura 4. *Resultados del clustering para k=8. Fuente: elaboración propia*

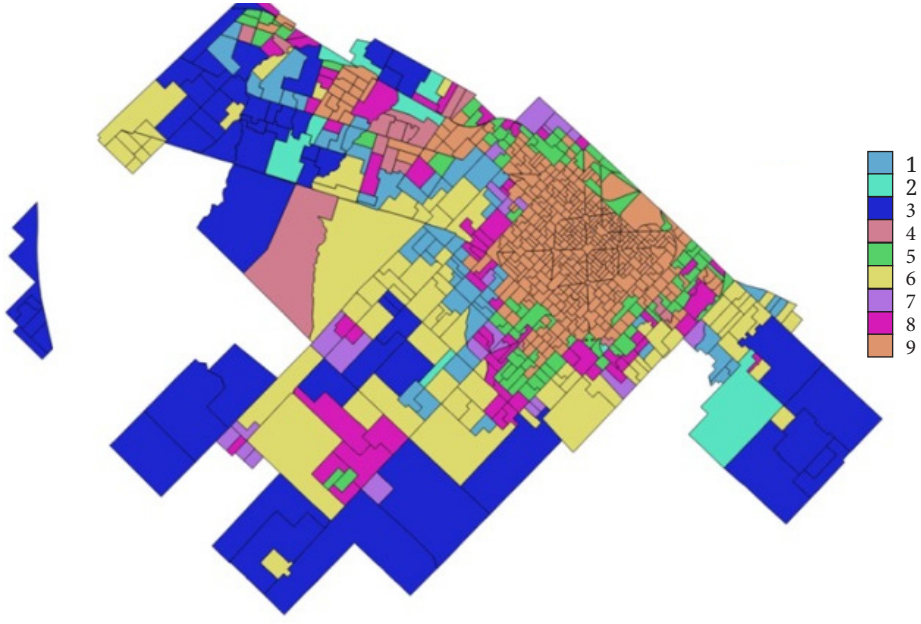

Figura 4.1. *Resultados del clustering para k=9. Fuente: elaboración propia*

Tabla 9

Coordenadas de los centroides para k = 10 clases

% viviendas con saneamiento cloacal por red	% viviendas con gas por red	% viviendas con agua potable por red	% viviendas con electricidad por red	Número de clase	Color
64,8	42,24	97,28	94,2	1	
96,57971	92,166667	97,775362	98,92029	2	
22,1	76,75	20,15	98,55	3	
88,1	74,5	98,816667	97,166667	4	
7,804878	7,804878	90,560976	89,195122	5	
61,8125	77,46875	94,15625	97,90625	6	
12	71,555556	95,155556	97,222222	7	
99,515957	98,577128	99,558511	99,335106	8	
3,6097561	14,390244	8,1219512	91,902439	9	
5,0425532	38,191489	92,87234	93	10	

Fuente: elaboración propia

Si se continúa aumentando el número de clases, al considerar 10 clases la heterogeneidad (discontinuidad) en la periferia persiste y vuelven a aparecer discontinuidades espaciales (radios censales color rojo) en la parte urbana consolidada (radios censales color marrón claro). Se observa nuevamente una significativa proximidad entre las coordenadas de los centroides de las clases 2 y 8 (tabla 9), lo que explica la aparición de algunos radios censales (rojos) dentro del área urbana consolidada (radios censales color marrón claro).

Por último, y a los efectos de establecer el número apropiado de clases, se puede observar en el diagrama de Elbow (figura 6) cómo disminuye la suma de las distancias al cuadrado (de cada radio censal con su centroide más cercano) en función del número de clases. Para ello se calculan, para valores de k entre 2 y 10, las distancias al cuadrado de los radios censales a su correspondiente centroide, utilizando la fórmula de la distancia euclídea, en este caso, considerando que cada radio censal es un vector 4-dimensional (considerando los cuatro servicios analizados).

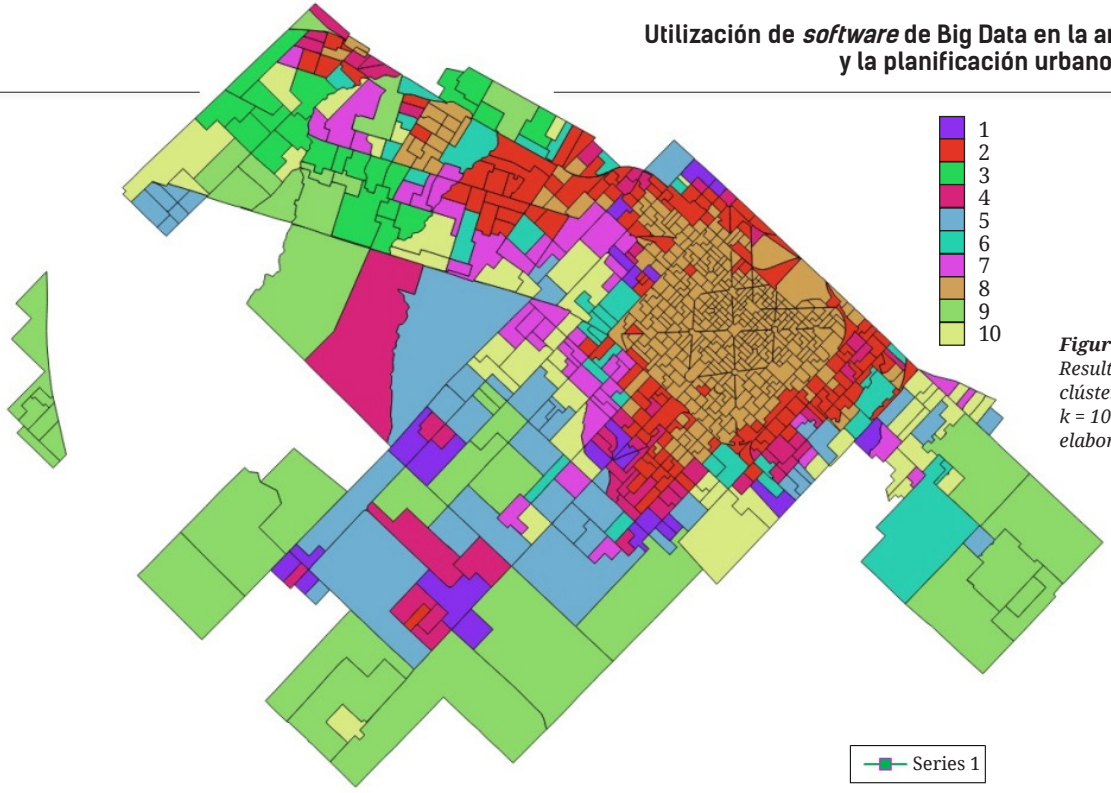

***Figura 5**. Resultados del clústering para k = 10. Fuente: elaboración propia*

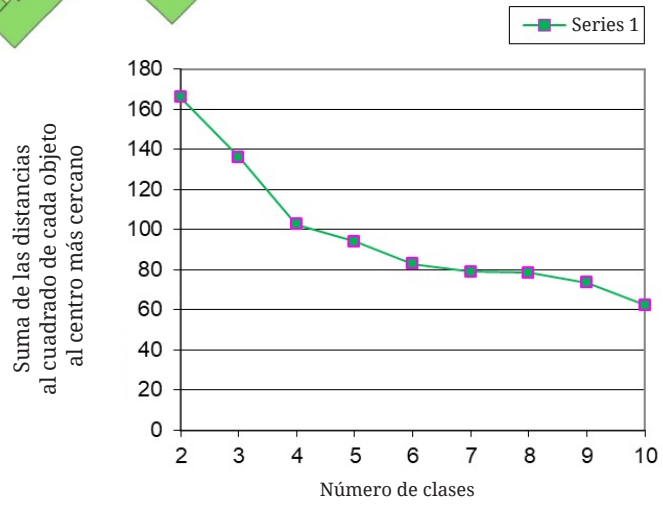

***Figura 6**. Diagrama de Elbow donde se observa la suma de las distancias al cuadrado de cada punto al centroide de su clase, en función del número de clases. Fuente: elaboración propia*

A partir de la información deducida de las figuras 1 a 5 y del diagrama de la figura 6 se puede justificar la adopción de un número "óptimo" de clases que agrupen a radios censales con similares porcentajes de cobertura de sus cuatro servicios urbanos básicos considerados y que presenten, asimismo, continuidad espacial. Por lo expuesto anteriormente, y considerando el objetivo de la clasificación en cuanto a detectar áreas homogéneas de prestación de servicios y en lo que respecta a la continuidad desde el punto de vista espacial, se sugiere considerar seis clases para realizar los correspondientes mapas de cobertura de servicios. Además, si se cuenta con información acerca del número de viviendas y habitantes por cada radio censal, se podría estimar el número de viviendas y de habitantes que carecen de cada servicio para, de esta manera, analizar la viabilidad de aumentar la cobertura de los servicios y sus correspondientes beneficios para los habitantes del lugar.

Conclusiones

Como conclusión general puede decirse que el trabajo realizado permitió identificar, agrupar y localizar con precisión las áreas urbanas con similares características (en cuanto a prestación de servicios), visibilizando vacancias en las coberturas de cada uno de los servicios. Por lo tanto, si por cada radio censal se cuenta con información acerca del número de viviendas, es posible deducir cuántas de ellas carecen de cada uno de los servicios y, considerando que ampliar la red de estos tiene un costo diferente por cada servicio, se puede estimar el costo que implicaría en términos económicos mejorar la situación observada. Por lo anterior, se deduce que tales datos pueden servir para detectar áreas con carencias de servicios, así como servir de soporte para la toma de decisiones a la hora de adoptar políticas o medidas de intervención para mejorar la habitabilidad de los lugares más desfavorecidos, y también analizar posibles intervenciones para realizar en lugares con similares prestaciones, estimar el precio de viviendas a partir de conocer el número de servicios disponibles e inferir posibles lugares de la periferia hacia donde puede crecer la ciudad, entre otras posibles aplicaciones.

En cuanto a los resultados obtenidos al aplicar el algoritmo de agrupamiento, puede decirse que si bien el diagrama de Elbow brinda información valiosa en relación con la distancia promedio de los puntos a su correspondiente centroide, el número K de clases "ideal", en realidad, dependerá del tipo de resultado que se busca. Así, por ejemplo, si K es igual al número de elementos para clasificar, cada clase tendrá un solo elemento y la distancia promedio

al centroide será 0. Sin embargo, por más deseable que sea esta última característica, el resultado que se obtendría sería el mismo que si no se realizara ninguna clasificación. Asimismo, si el número de clases que encontrar fuese igual a 1, entonces todos los elementos formarían parte de la misma (y única) clase, por lo que la clasificación no tendría ningún sentido. Por lo tanto, el número de clases K deberá ser estrictamente mayor a 1 y menor a la cantidad de elementos para clasificar. En esta instancia, será el especialista en el dominio del problema quien determine (justificadamente) la cantidad "correcta" de clases en las que se pueden agrupar los elementos para clasificar. En este trabajo, el resultado obtenido para k = 6 es considerado el ideal para el caso de estudio, pues se trata de un número razonable de grupos homogéneos y que además presentan continuidad espacial. Las coordenadas de los centroides de cada una de las seis clases describen un porcentaje de cobertura de cada servicio, y cada radio censal se asigna al centroide más cercano.

Con respecto al *software* y *hardware* utilizados para procesar Big Data, se pudo comprobar que el sistema de archivos distribuido (HDFS) de Hadoop permite agrupar en un mismo lugar datos procedentes de diversos equipos que componen un clúster de PC. De esa forma, se logra una mayor capacidad de almacenamiento de información que la que se tendría usando un solo equipo. Esto es necesario cuando el tamaño de los datos para procesar excede la capacidad de almacenamiento del equipo de un usuario. En este tipo de situaciones es viable utilizar el *framework* Hadoop y, sobre él, correr aplicaciones escritas para Apache Spark para aprovechar, por un lado, la capacidad de almacenamiento distribuido del HDFS de Hadoop y, por otro, la velocidad de procesamiento de grandes volúmenes de datos que ofrece Apache Spark. De este modo, es posible incrementar el volumen de datos sin requerir ningún cambio en la aplicación desarrollada. En caso de que se necesite agregar nuevos nodos a un clúster o balancear la carga de datos para procesar en cada nodo, será el *framework*, en este caso Apache Spark, el encargado de realizar esta tarea permitiendo al programador enfocarse en encontrar una solución al problema que pretende resolver.

Referencias bibliográficas

APACHE SOFTWARE FOUNDATION (2019a). Apache Spark. URL: https://spark.apache.org/. Accedido: 30-7-2019.

APACHE SOFTWARE FOUNDATION (2019b). Apache Spark MLlib. URL: https://spark.apache.org/mllib/. Accedido: 30-7-2019.

HERNÁNDEZ ORALLO, José; RAMÍREZ QUINTANA, María. José y FERRI RAMÍREZ, César. (2004). *Introducción a la minería de datos*. Madrid: Pearson.

INDEC. (2010). Censo nacional de población, hogares y viviendas 2010. Recuperado de: https://www.indec.gob.ar/indec/web/Nivel4-Tema-2-41-135

KOSELEVA, Natalija & ROPAITE, Guoda (2017). Big data in building energy efficiency: understanding of big data and main challenges. *Procedia Engineering 172*. pp. 544-549.

MAC QUEEN, James B. (1967). Some methods for classification and analysis of multivariate observations. In: Proc. *5th Berkeley Symposium on mathematical statistics and probability*. 1: 281-297. University of California Press.

SCHINTLER, Laurie & CHEN, Zhenhua (Eds.). (2018). *Big data for regional science*. Routledge.

THAKURIAH, Piyushimita; TILAHUN, Nebiyou & ZELLNER, Moira (Eds.). (2017). *Seeing cities through Big Data. Research, methods and applications in urban informatics*. Springer.

WANG, Stephen Jia & MORIARTY, Patrick (2018). *Big Data for urban sustainability: A human-centered perspective*. Springer.

WITTEN, Ian H. & FRANK, Eibe *(2000). Data Mining: Practical Machine Learning. Tools and techniques with Java implementations.* Morgan Kaufmann Publishers.

WU, Xindong; KUMAR, Vipin; ROSS QUINLAN, J.; GHOSH, Joydeep; YANG, Qiang; MOTODA, Hiroshi.... STEINBERG, Dan. (2008). Top-10 algorithms in data mining. *Journal Knowledge and information systems*. *Vol. 14*. Issue 1. pp. 1-37. Springer.

ZHOU, Kaile & YANG, Shanlin (2016). Understanding household energy consumption behaviour: The contribution of energy big data analytics. *Renewable and Sustainable Energy Reviews*. Vol. 56. pp. 810-819.