



UNIVERSIDAD DE CASTILLA-LA MANCHA

ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

**TECNOLOGÍA ESPECÍFICA DE
COMPUTACIÓN**

TRABAJO FIN DE GRADO

Sistema para la captura de datos de usuarios e inventario de red de
una empresa de gestión de la red de distribución de agua potable

Francisco Martínez Esteso

2 de Diciembre de 2019





UNIVERSIDAD DE CASTILLA-LA MANCHA

ESCUELA SUPERIOR DE INGENIERÍA INFORMÁTICA

GRADO EN INGENIERÍA INFORMÁTICA

TECNOLOGÍA ESPECÍFICA DE
COMPUTACIÓN

TRABAJO FIN DE GRADO

Sistema para la captura de datos de usuarios e inventario de red de
una empresa de gestión de la red de distribución de agua potable

Autor: Francisco Martínez Esteso

Directores: Elena María Navarro Martínez

Juan Peralta Malvar

2 de Diciembre de 2019

“Crea y actúa como si fuera imposible fallar.”

Charles F. Kettering.

Declaración de Autoría

Yo, Francisco Martínez Esteso con DNI 48260191G, declaro que soy el único autor del Trabajo Fin de Grado titulado *Sistema para la captura de datos de usuarios e inventario de red de una empresa de gestión de la red de distribución de agua potable*, y que el citado trabajo no infringe las leyes en vigor sobre propiedad intelectual y que todo el material no original contenido en dicho trabajo está apropiadamente atribuido a sus legítimos autores.

Albacete, a 2 de Diciembre de 2019

Fdo.: Francisco Martínez Esteso

Resumen

Los Sistemas de Información Geográfica (SIG) se han convertido en una herramienta fundamental en el día a día de un profesional de las tecnologías de la información enrolado en el área del geoprocesamiento, pero también se hacen imprescindibles en otros campos profesionales como son la cartografía, la agricultura y el sector transporte, entre otros muchos. También cada día son más las empresas que deciden embarcarse en proyectos de Transformación Digital de sus organizaciones, buscando incrementar la productividad y eficiencia de sus compañías a través del uso de las tecnologías SIG. Para ello crean diferentes tipos de infraestructuras que buscan mejorar las capacidades geoespaciales de la empresa en diferentes aspectos como, por ejemplo, la mejora de las interacciones entre clientes y usuarios. El aumento de la automatización en los procesos empresariales y el auge de las nuevas técnicas de inteligencia artificial han permitido a los Sistemas de Información Geográfica enfrentarse a nuevos retos, trabajando en áreas donde nunca antes había tenido cabida este tipo de sistemas. A todo ello se añade la mejora de los SIG con respecto al análisis de datos y exploración de patrones de la información geográfica.

Este proyecto muestra el proceso de desarrollo llevado a cabo en un proyecto SIG real, en el que se realiza un proceso de catastro técnico y comercial para una empresa de aguas peruana ubicada en la ciudad de Chimbote. El proyecto se ha llevado a cabo íntegramente en el seno de una empresa española, Eptisa, con amplia experiencia en trabajos relacionados con la información geográfica tanto en España como en el resto del mundo.

En esta memoria se documentan los trabajos llevados a cabo en colaboración con un equipo de profesionales internacionales, cuyo fin es la codificación de una cartografía mediante la ejecución de procesos automáticos en segundo plano y la creación de un sistema de clasificación de elementos cartográficos que da apoyo al equipo técnico en las tareas de digitalización de las parcelas

catastrales de la cartografía base. Además, el proyecto termina resaltando la necesidad de crear diferentes herramientas de gestión y tratamiento geográfico de datos para la remodelación de diferentes elementos de la cartografía.

Agradecimientos

Me gustaría empezar agradeciendo el trato y dedicación de mi tutora académica Dña. Elena María Navarro Martínez. Gracias por apoyarme, pero sobre todo por ayudarme en la realización de este proyecto y en la resolución de todas las dudas que me han surgido durante el periodo de realización de este trabajo.

Seguidamente, me gustaría agradecer a D. Juan Peralta Malvar, tutor profesional de este proyecto, por la ayuda teórica, técnica e incluso moral, que me ha brindado durante todos estos meses de proyecto. Gracias por enseñarme, por hacerme mejor profesional, por darme la oportunidad de seguir aprendiendo y, sobre todo, por mostrarme la pasión y dedicación incansable que le dedicas a tu trabajo día a día, esa con la que tantas veces me has dejado impresionado.

No me gustaría olvidarme de mis compañeros de oficina, en los que tantas veces he volcado mis dudas y, tantas otras, he obtenido la respuesta precisa que andaba buscando. En especial me gustaría nombrar a mis dos grandes mentores profesionales, Dña. María José Montoya Martínez y D. Sergio Salmerón González, compañeros de trabajo y amigos, a los que no podré devolver nunca la transferencia de saberes y entrega que me han brindado siempre. Para mí ha sido un placer trabajar con vosotros, y una verdadera suerte haberos conocido. Gracias.

También me gustaría agradecer el trabajo de corrección y uso lingüístico que han realizado D. Francisco Martínez Montoya y D. Pedro Pardo Arias, los cuales han sumado su tiempo y su saber a la realización de este proyecto de la manera más eficaz y desinteresada posible.

Por último, agradecer a mi familia, y especialmente a mi compañera de vida, por el apoyo incondicional que me habéis dado, porque, gracias a vosotros, hoy nada es imposible.

Índice

Capítulo 1	Introducción	1
1.1	Introducción	1
1.2	Objetivos del TFG	2
1.3	Estructura de la memoria	2
Capítulo 2	Antecedentes, estado de la cuestión.....	5
2.1	Introducción	5
2.2	Introducción a los Sistemas de Información Geográfica.....	5
2.2.1	¿Qué es un SIG?.....	5
2.3	Componentes de un SIG	6
2.3.1	Historia de los SIG	8
2.3.2	Herramientas para el desarrollo de los SIG	12
2.4	Introducción a Machine Learning	15
2.4.1	Vocabulario básico	17
2.4.2	Tipos de Aprendizaje Automático.....	18
2.4.3	Proceso de construcción de modelos	19
2.5	Conclusiones	20
Capítulo 3	Contexto y Metodología	21
3.1	Introducción	21
3.2	Contexto del TFG	21
3.3	Fases del proyecto	23
3.4	Conclusiones	24
Capítulo 4	Codificación de la cartografía base	25
4.1	Introducción	25
4.2	Convenio de visualización de la cartografía base	25
4.3	Cartografía base.....	26
4.4	Estudio de la tecnología.....	35
4.5	Codificación de subsectores	37
4.6	Numeración de manzanas.....	37
4.6.1	Codificar manzanas.....	38

4.6.2	Algoritmo de numeración de manzanas	39
4.7	Numeración de esquinas.....	43
4.7.1	Normalización de elementos de la cartografía base	44
4.7.2	Algoritmo de numeración de esquinas	45
4.8	Numeración de lotes.....	46
4.8.1	Codificación de lotes.....	47
4.8.2	Algoritmo de numeración de lotes	47
4.9	Utilidades y mejoras	50
4.10	Resultados	52
Capítulo 5	Sistema de clasificación de elementos cartográficos	55
5.1	Introducción	55
5.2	Estudio de necesidades	55
5.3	Construcción de la base de conocimiento.....	59
5.4	Algoritmo.....	61
5.5	Construcción del sistema.....	64
5.5.1	Análisis de alternativas	64
5.5.2	Implementación de la solución	66
5.6	Resultados	71
5.7	Conclusiones	78
Capítulo 6	Conclusiones y trabajos futuros	79
6.1	Conclusiones	79
6.2	Trabajos futuros y mejoras	81
6.3	Valoración personal del trabajo realizado.....	82
6.4	Competencias desarrolladas	83

Listado de figuras

FIGURA 1. ESQUEMA DE REPRESENTACIÓN DEL MODELO CLÁSICO DE UN SIG (VÍCTOR OLAYA, 2006).	7
FIGURA 2. INFOGRAFÍA TEMPORAL DE LA HISTORIA DE LOS SIG ACTUALES.	9
FIGURA 3. INFOGRAFÍA DE LOS DATOS PRODUCIDOS POR UN USUARIO CADA MINUTO.	16
FIGURA 4. DIAGRAMA DE GANTT QUE MUESTRA LA PLANIFICACIÓN DEL TIEMPO DEL PROYECTO.	24
FIGURA 5. IMAGEN PREVIA DE LA CAPA SECTORES.	28
FIGURA 6. IMAGEN PREVIA DE LA CAPA SUBSECTORES.	29
FIGURA 7. IMAGEN PREVIA DE LA CAPA MANZANAS.	30
FIGURA 8. IMAGEN PREVIA DE LA CAPA LOTES.	31
FIGURA 9. IMAGEN PREVIA DE LA CAPA ZONA PILOTO.	32
FIGURA 10. IMAGEN PREVIA DE LA CAPA PUEBLOS.	33
FIGURA 11. IMAGEN PREVIA DE LA CARTOGRAFÍA BASE.	34
FIGURA 12. REQUISITOS DE NUMERACIÓN DE MANZANAS.	38
FIGURA 13. BANDA HORIZONTAL DE APOYO PARA LA NUMERACIÓN DE MANZANAS.	40
FIGURA 14. SEGUNDA ITERACIÓN DEL ALGORITMO DE NUMERACIÓN DE MANZANAS.	42
FIGURA 15. REPRESENTACIÓN VISUAL DEL RESULTADO DEL AVANCE DEL ALGORITMO DE NUMERACIÓN.	43
FIGURA 16. REQUISITOS DE NUMERACIÓN DE ESQUINAS.	44
FIGURA 17. VÉRTICES MAL CODIFICADOS EN MANZANAS DE LA CARTOGRAFÍA.	45
FIGURA 18. REQUISITOS DE NUMERACIÓN DE LOTES.	47
FIGURA 19. VÉRTICES DE LOTES QUE INTERSECAN CON LA SILUETA DE LA MANZANA.	48
FIGURA 20. VÉRTICES DE LOTES QUE INTERSECAN CON EL PERÍMETRO DE LA MANZANA.	49
FIGURA 21. SALIDA POR CONSOLA DE LA EJECUCIÓN DEL SCRIPT DE NUMERACIÓN DE SUBSECTORES.	51
FIGURA 22. MANZANAS NUMERADAS TRAS EL PROCESO DE CODIFICACIÓN DE LA CARTOGRAFÍA BASE.	52
FIGURA 23. ESQUINAS NUMERADAS TRAS EL PROCESO DE CODIFICACIÓN DE LA CARTOGRAFÍA BASE.	53
FIGURA 24. LOTES NUMERADOS TRAS EL PROCESO DE CODIFICACIÓN DE LA CARTOGRAFÍA BASE.	53
FIGURA 25. CHEAT SHEET CREADO POR SCIKIT LEARN.	62
FIGURA 26. CHEAT SHEET CREADO POR DLIB.	63
FIGURA 27. CREACIÓN DE UN NUEVO PROYECTO EN MAMLS.	66
FIGURA 28. BASE DE CONOCIMIENTO CREADA EN MAMLS.	67
FIGURA 29. ESQUEMA DEL EXPERIMENTO DE CLUSTERING EN MAMLS.	68
FIGURA 30. MAPA TEMÁTICO DEL RESULTADO DE LA CLASIFICACIÓN DE LA CARTOGRAFÍA.	72
FIGURA 31. DISTRIBUCIÓN DE LA VARIABLE CLASE DE MANZANAS.	73
FIGURA 32. ESQUEMA DEL EXPERIMENTO TRAS AÑADIR EL MÓDULO DE EVALUACIÓN.	74
FIGURA 33. RESULTADOS DE EVALUACIÓN DEL MODELO TRATADO.	75
FIGURA 34. GRÁFICA DE BARRAS DE LA DISTRIBUCIÓN DE LA VARIABLE CLASE DE MANZANAS.	76
FIGURA 35. DISTRIBUCIÓN DE LA CANTIDAD DE LOTES EN EL LADO C2 DE LAS MANZANAS.	76
FIGURA 36. GRÁFICO QUE MUESTRA LOS CLÚSTERES DE MANZANAS CREADOS.	77
FIGURA 37. VISTA EXPLORATORIA PREVIA A LA EJECUCIÓN DE LA TOOL PARA MANZANAS TIPO 1.	87

FIGURA 38. VISTA EXPLORATORIA TRAS LA EJECUCIÓN DE LA TOOL PARA MANZANAS TIPO 1.....	87
FIGURA 39. VISTA EXPLORATORIA PREVIA A LA EJECUCIÓN DE LA TOOL PARA MANZANAS TIPO 2.	90
FIGURA 40. VISTA EXPLORATORIA TRAS LA EJECUCIÓN DE LA TOOL PARA MANZANAS TIPO 2.....	90
FIGURA 41. VISTA EXPLORATORIA PREVIA A LA EJECUCIÓN DE LA TOOL PARA MANZANAS TIPO 3_A.	93
FIGURA 42. VISTA EXPLORATORIA TRAS LA EJECUCIÓN DE LA TOOL PARA MANZANAS TIPO 3_A.	93
FIGURA 43. VISTA EXPLORATORIA PREVIA A LA EJECUCIÓN DE LA TOOL PARA MANZANAS TIPO 3_B.	96
FIGURA 44. VISTA EXPLORATORIA TRAS LA EJECUCIÓN DE LA TOOL PARA MANZANAS TIPO 3_B.	96

Listado de tablas

TABLA 1. DESCRIPCIÓN DE LA PLATAFORMA CARTO	12
TABLA 2. DESCRIPCIÓN DE LA PLATAFORMA MAPBOX.....	12
TABLA 3. DESCRIPCIÓN DEL SISTEMA MAPINFO.	13
TABLA 4. DESCRIPCIÓN DEL SISTEMA SMALLWORLD.	14
TABLA 5. PLANIFICACIÓN DEL TIEMPO DEL PROYECTO EN ETAPAS DE DESARROLLO.....	23
TABLA 6. CONVENIO DE COLORES PARA LA VISUALIZACIÓN DE LA CARTOGRAFÍA.....	26
TABLA 7. DEFINICIÓN DE LA CAPA SECTORES.	28
TABLA 8. DEFINICIÓN DE LA CAPA SUBSECTORES.	29
TABLA 9. DEFINICIÓN DE LA CAPA MANZANAS.....	30
TABLA 10. DEFINICIÓN DE LA CAPA LOTES.	31
TABLA 11. DEFINICIÓN DE LA CAPA ZONA PILOTO.	32
TABLA 12. DEFINICIÓN DE LA CAPA PUEBLOS.	33
TABLA 13. DEFINICIÓN DE LA CARTOGRAFÍA BASE.....	34
TABLA 14. LENGUAJES DE PROGRAMACIÓN INTEGRADOS POR ARCGIS PARA DESARROLLADORES.....	36
TABLA 15. ESTUDIO DE NECESIDADES DE LA BASE DE DATOS.	56
TABLA 16. TIPOS DE MANZANAS PREDEFINIDAS POR OBSERVACIÓN DIRECTA.....	58
TABLA 17. DEFINICIÓN DE VARIABLES DE LA BASE DE CONOCIMIENTO.....	60
TABLA 18. COMPARATIVA DE SERVICIOS MACHINE LEARNING EN LA NUBE.	65
TABLA 19. CONFIGURACIÓN PARA LA CREACIÓN DEL MODELO MAMLS.	69
TABLA 20. VARIABLES DEL DATASET DE SALIDA TRAS EJECUTAR EL ALGORITMO DE CLUSTERING.....	70
TABLA 21. LEYENDA DEL MAPA TEMÁTICA QUE MUESTRA LOS RESULTADOS DE LA CLASIFICACIÓN.....	71
TABLA 22. ANÁLISIS DE RESULTADOS DEL SISTEMA DE CLASIFICACIÓN.	72
TABLA 23. COMPETENCIAS DESARROLLADAS EN EL PROYECTO.....	83
TABLA 24. DESCRIPCIÓN DE LA HERRAMIENTA DE REMODELACIÓN DE LOTES DE MANZANAS DE TIPO 1.....	86
TABLA 25. DESCRIPCIÓN DE LA HERRAMIENTA DE REMODELACIÓN DE LOTES DE MANZANAS DE TIPO 2.....	88
TABLA 26. DESCRIPCIÓN DE LA HERRAMIENTA DE REMODELACIÓN DE LOTES DE MANZANAS DE TIPO 3_A.	91
TABLA 27. DESCRIPCIÓN DE LA HERRAMIENTA DE REMODELACIÓN DE LOTES DE MANZANAS DE TIPO 3_B.....	94

Capítulo 1

Introducción

1.1 Introducción

El principal activo de una empresa de gestión de red de agua es la red de distribución que le permite dar el servicio a los clientes. Desde el punto de vista de los sistemas de información que dan soporte a estas empresas, la red de distribución de agua tiene algunas características especiales, entre las que destaca la componente espacial de los datos, es decir, cómo está distribuida espacialmente en el territorio. La distribución espacial de la red de agua hace necesario disponer de un elemento de referencia para ubicarla (Arreguin, 2015). Este elemento es la cartografía, la cual permite dotar de contexto a la información de los activos de la red.

Elaborar una cartografía base es un proceso complejo y costoso en el tiempo (Osorio Campillo & Sánchez, 2011), en el que suelen participar equipos multidisciplinarios, y parte de ese proceso es la codificación de forma única de los elementos. Por ejemplo, las parcelas catastrales (o lotes en terminología latinoamericana) siempre tienen un código único que permite identificar la zona en la que se encuentran. Hoy en día, las empresas buscan reducir al máximo el tiempo de desarrollo de los proyectos, a fin de conseguir la máxima rentabilidad, reduciendo los recursos dedicados a cada tarea por unidad de tiempo. Esto hace que la automatización de estas tareas, en todas las áreas de la informática, sea clave a la hora de conseguir los objetivos de cualquier proyecto en el menor periodo de tiempo. En este proyecto se busca automatizar muchas de las tareas

que manualmente podría realizar un técnico en Sistemas de Información, pero que, debido al volumen de los datos inherentes a una cartografía municipal, es muy costosa en tiempo. Por ello, y a través del uso de tecnologías adecuadas, asumiendo el coste temporal que conlleva el necesario análisis, se ha trabajado en reducir los costes temporales relacionados con algunos de los procesos de producción cartográfica, mediante la automatización de tareas apoyándonos en los Sistemas de Información Geográfica (SIG) para ello.

1.2 Objetivos del TFG

El objetivo del trabajo es diseñar e implementar un sistema para preparar la cartografía base de los municipios de Chimbote y Nuevo Chimbote de forma que esta permita la realización del trabajo de campo del catastro comercial y técnico. Dicho objetivo se desglosa en los siguientes subobjetivos:

- OBJ 1. Estudiar y realizar un análisis de los Sistemas de Información Geográfica. Introducción y puesta en contexto de estos para un mejor conocimiento de sus funciones y estructura.
- OBJ 2. Estudiar y realizar un análisis de las diferentes técnicas de Machine Learning existentes y que son candidatas de formar una solución del problema tratado.
- OBJ 3. Realizar un estudio de necesidades para valorar las tecnologías que mejor se adaptan a los requisitos impuestos por el cliente en el proceso de construcción de la solución final.
- OBJ 4. Implementar un sistema de codificación de la cartografía base para garantizar la unicidad de los elementos cartográficos y la asignación de dichos códigos mediante una serie de reglas espaciales definidas a priori.
- OBJ 5. Implementar un sistema para automatizar la producción cartográfica de algunos elementos (lotes) a partir de su morfología y relaciones espaciales.

1.3 Estructura de la memoria

A continuación, se especifica la organización de cada una de las partes de la memoria de este proyecto:

- **Capítulo 1: Introducción.** Este primer capítulo ha descrito el contexto del presente trabajo. Además, describe la estructura que sigue la presente memoria.
- **Capítulo 2: Antecedentes, estado de la cuestión.** Se sigue el desarrollo de la memoria con una puesta del contexto de nuestra solución, así como un análisis de los Sistemas de Información Geográfica, así como de las propuestas de Machine Learning.
- **Capítulo 3: Contexto y Metodología.** En este capítulo se presentan los métodos de trabajo adoptados, definiendo también las distintas fases que se han abordado durante el desarrollo de este proyecto y las convenciones utilizadas para facilitar la lectura del resto de la memoria.
- **Capítulo 4: Codificación de la cartografía base.** En este capítulo se describe la implementación realizada de los procesos automáticos de codificación de la cartografía base disponible.
- **Capítulo 5: Sistema de clasificación de elementos cartográficos.** En este capítulo se describe el sistema de aprendizaje desarrollado para clasificar los diferentes tipos de elementos cartográficos.
- **Capítulo 6: Conclusiones y trabajos futuros.** Para terminar, se realiza un estudio y valoración de los resultados obtenidos frente a los esperados al inicio de este proyecto, poniendo en valor los objetivos perseguidos y el resultado final. También se valoran futuras líneas de trabajo y posibles mejoras de nuestro trabajo.

Capítulo 2

Antecedentes, estado de la cuestión

2.1 Introducción

En este capítulo se busca introducir al lector al mundo de los Sistemas de Información Geográfica, dando una primera visión de los mismos, definiendo sus componentes y estructura. También se hace una presentación de los algoritmos de aprendizaje automático, repasando sus características y definiendo algunos términos básicos relacionados.

2.2 Introducción a los Sistemas de Información Geográfica

El objetivo de esta sección es poner en contexto los próximos capítulos, así como acercar al lector el concepto de los Sistemas de Información Geográfica (SIG en español o GIS en inglés).

2.2.1 ¿Qué es un SIG?

Un SIG es una herramienta válida para el análisis, recopilación y gestión de datos espaciales. Tiene sus raíces en las ciencias geográficas, integrando gran cantidad de tipos de datos. Son capaces de analizar y trabajar con la ubicación espacial, ofreciendo visualizaciones en forma de mapas, incluso en 3D (Maguire, 1991). Una definición más formal la presenta (Víctor Olaya, 2006), quien lo define como “un sistema que integra tecnología informática, personas e información geográfica, y cuya principal función es capturar, analizar, almacenar, editar y representar datos georreferenciados”.

Los SIG permiten la digitalización y automatización de las actividades propias de la cartografía convencional. Con este potencial a nuestro alcance, podemos hacer un análisis más profundo de los datos, descubriendo nuevos patrones invisibles a simple vista y extrayendo conclusiones más complejas.

La tecnología GIS combina el potencial de las tecnologías de la información con la ciencia de la cartografía, en busca de un objetivo común: Obtener conocimiento de todos los tipos de datos disponibles. Esta unión de diferentes ramas de la ciencia, como son la cartografía y la informática entre otras, en busca de resolución de problemas complejos, ha dado lugar a una nueva disciplina que conocemos como Ciencia de la Información Geográfica (Víctor Olaya, 2006).

Por ejemplo, una empresa que brinda un servicio de gestión de aguas municipales puede registrar los datos espaciales de su red de abastecimiento de agua, mediante un Sistema de Información Geográfica pudiendo así gestionar los datos que tenga en su poder. También podrá generar nuevos datos mediante su análisis, así como obtener una representación posterior en un mapa que permita identificar patrones o visualizar la información obtenida de una manera más cómoda. De esta manera obtendrá un registro de todos los elementos de la red, como tuberías, codos, bocas de agua y alcantarillas entre otros, además de un conjunto muy amplio de información, ligada a esa red, como la longitud de la misma o su morfología.

2.3 Componentes de un SIG

Existen diferentes conceptos acerca de los componentes o elementos que conforman un sistema GIS. El modelo tradicional se compone de 5 componentes definidos que son (Víctor Olaya, 2006):

- **Datos:** Parte principal y materia prima necesaria de un SIG, es el elemento sobre el que gira todo el entorno de gestión de datos geográficos.
- **Métodos:** Conjunto de herramientas que pueden ser ejecutadas sobre los datos disponibles.
- **Software:** Sistema informático que permite la gestión de los datos.
- **Hardware:** Componentes computacionales sobre los que ejecutar el componente anterior.
- **Personas:** Usuario principal del modelo.

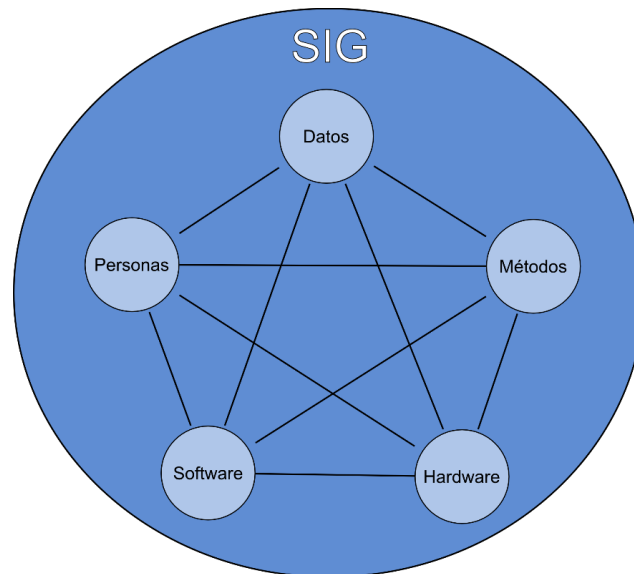


Figura 1. Esquema de representación del modelo clásico de un SIG (Víctor Olaya, 2006).

Sin embargo, este modelo ha ido evolucionando, adecuándose a las necesidades de los problemas y requisitos de los sistemas de hoy en día. Por ello resulta conveniente someter a estudio la evolución del sistema clásico y establecer nuevos elementos, como son (Víctor Olaya, 2006):

- **Datos:** Ya formaba parte del sistema clásico, ya que es una de las razones de la existencia de los SIG. Estos son cada día más abundantes y complejos, por ello se debe contemplar también en este apartado el procesamiento que se hace de ellos, el tratamiento y la calidad de los mismos. También es importante tener en cuenta cómo se almacenan dichos datos.
- **Análisis:** Los Sistemas de Información Geográfica cuentan con una batería de formulaciones que permiten automatizar procesos de tratamiento de datos, no siendo exclusivamente procesos cartográficos. Esta es una de las funciones principales de los SIG y permite la automatización de tareas y la creación de nuevos procesos.
- **Visualización:** La información geográfica está ligada, inconscientemente, a una representación visual de ella, ya que es a lo que los consumidores finales de esa información están acostumbrados. Además, permite el descubrimiento de nuevos patrones en el análisis y una mejor interpretación. Un SIG, por tanto, debe incluir la capacidad de representación gráfica de los datos numéricos que almacena.
- También es importante apuntar, que un SIG no debe olvidar la metodología clásica de impresión de mapas y debe ofrecer dicha funcionalidad.

- **Tecnología:** Se incluye en este aspecto tanto la parte física, o *hardware*, como la parte virtual, o *software*, de los SIG. En el área física, podemos encontrar los periféricos que proporcionan la información de entrada, y los sistemas de impresión. Por otro lado, el *software* es la parte que se encarga de gestionar y analizar los datos. Este ha sufrido un gran avance en los últimos tiempos, pasando de un modelo tradicional de visualización, gestión y análisis de los datos geográficos, a otro modelo en el que cada una de estas tareas es llevada a cabo en un entorno o sistema distinto, especializando así el software relativo para cada proceso.
- **Factor organizativo:** Los sistemas SIG necesitan de una relación entre las personas que hacen uso del mismo, así como de relación entre los componentes que lo integran. Por ello es cada vez más frecuente la estandarización y normalización de sus elementos. Esto se debe al auge de metodologías de trabajo en las que los datos no son propiedad del usuario, sino que se comparten como un recurso global con el que trabajar. Además, un gran número de perfiles profesionales muy variados toman contacto con dichos elementos. Entre ellos se encuentra el usuario final, el cual participa en el proceso de análisis de esos datos, aportando nuevos puntos de vista e identificando patrones útiles para su comprensión.

2.3.1 Historia de los SIG

Los Sistemas de Información Geográfica (SIG) tienen sus primeros avances a partir de la década de los sesenta (Víctor Olaya, 2006), cuando los nuevos desarrollos ofrecidos por la informática y la necesidad creciente de gestionar la información geográfica permiten su aparición, ofreciendo un conjunto limitado de características y con una estructura característica que se mantiene todavía hoy en día.

El primer sistema de información geográfica conocido es el desarrollado por Roger Tomlinson en Canadá a principios de los años 60 (Víctor Olaya, 2006). La herramienta creada por Tomlinson permitía gestionar el inventario geográfico canadiense, dotando al usuario de una herramienta de análisis de la información del país. Tomlinson es considerado y llamado “*padre de los SIG*”.

Tras la aparición del sistema de Tomlinson se abre un abanico de necesidades en torno al almacenamiento y gestión de los datos geográficos nunca visto, por lo que proliferan descubrimientos y avances en el campo de los SIG en los años siguientes. Uno de ellos fue la aparición en 1964 de la aplicación SYMAP (Soderlund, Nelson, Shoemaker, & Paterson, 2006), desarrollada en el centro Harvard Laboratory. Ésta permitía la entrada de información en forma de figuras geométricas tales como puntos o líneas, lo cual es lo que actualmente conocemos como *información vectorial*. Tras ello, en 1969, David Sinton, también perteneciente a Harvard Laboratory, desarrolla GRID (Víctor Olaya, 2006), preámbulo de los Sistemas de Información Geográfica *ráster*, un programa que permitía utilizar la información en formato de matriz como entrada del sistema o almacenarla, lo cual no había sido posible hasta ese momento. También permitía almacenar los datos en este formato.

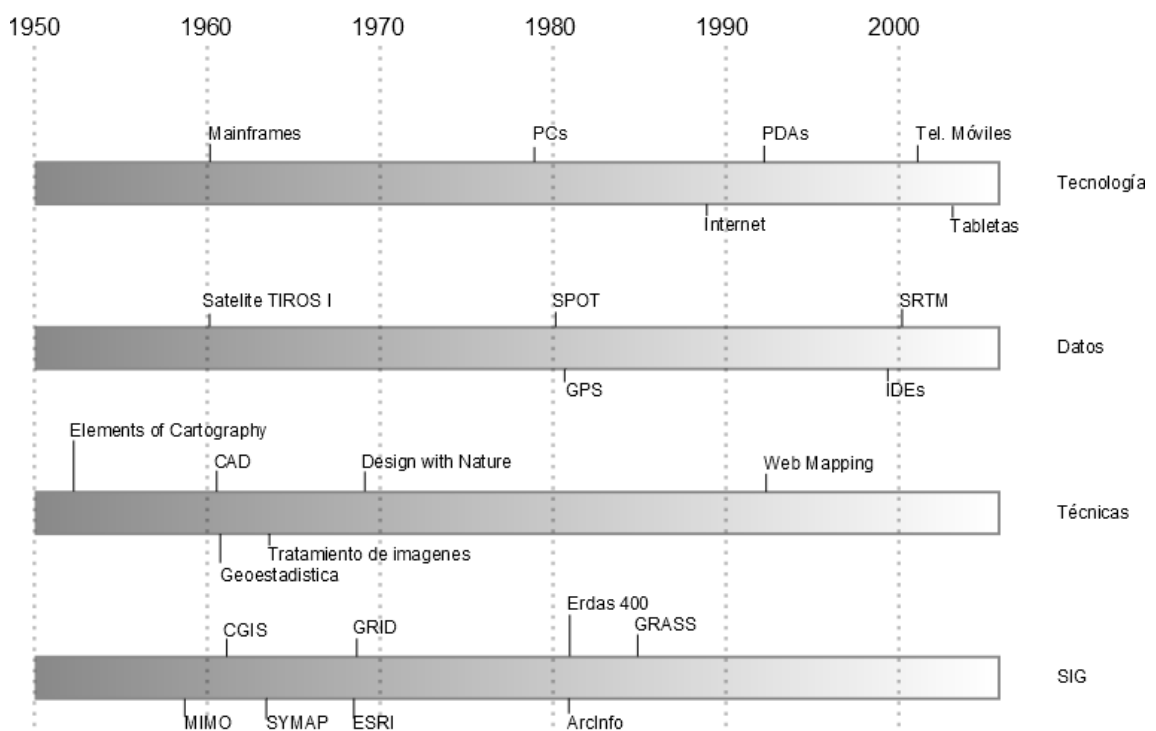


Figura 2. Infografía temporal de la historia de los SIG actuales.

Tras estos sistemas y ligados a una época de evolución histórica de los SIG, aparecen otros como SYMVU (Siabato, 2018), inicio de la representación tridimensional, CALFORM o IMGRID (Siabato, 2018), inspirados en el ya comentado sistema GRID, que incluye muchos de los elementos hoy necesarios para el análisis ráster. La Figura 2 muestra la evolución temporal de los diferentes sistemas.

Otro aspecto que caracteriza a los SIG es que han pasado de ser una herramienta de uso exclusivo de profesionales de las ciencias cartográficas, a englobar a un elenco de profesionales mucho mayor debido a la gran variedad de disciplinas que precisa. Así y conforme han ido pasando los años, los SIG han acaparado una gran atención, como demuestra la creación de diferentes listas de distribución, revistas, foros y eventos donde se discute y muestran los últimos avances del campo.

Desde el punto de vista tecnológico, las mejoras gráficas que han proliferado en la computación, así como el abaratamiento de los costes y la simplificación del almacenamiento, el acceso masivo de datos, así como los avances en la entrada de datos geográficos y su tratamiento han impulsado el desarrollo y mejora de los SIG. También el impacto de las tecnologías móviles ha sido determinante a la hora de difundir y mejorar estas tecnologías. Otros factores, como el impulso de las plataformas web o el desarrollo de los visores, también han ayudado al impulso de los SIG.

Desde el punto de vista de los datos, los SIG son totalmente dependientes de estos. Los primeros datos geográficos (de Fomento, 2009), fueron tomados a partir de las digitalizaciones de la cartografía impresa en papel. Posteriormente, se incorporaron las primeras imágenes terrestres (Víctor Olaya, 2006), en la época de la Primera Guerra Mundial, con fines militares y de defensa. A partir de entonces se producen varios hechos relevantes en cuanto a la toma de datos espaciales, como el primer lanzamiento de observación meteorológica, el *TIROS I*, en 1960 (Fritz & Wexler, 1960). Posteriormente, se lanzó en 1974 el satélite *KOSMOS* (Konecny, 2004). En 1980, se fundó *SPOT* (Víctor Olaya, 2006), la primera compañía que ofrecía imágenes de satélite procedentes de satélites propios para su venta comercial.

Otras de las tecnologías que irrumpen en favor de los datos y su precisión son el GPS, tecnología de posicionamiento (Víctor Olaya, 2006)(Correia, 2002) y localización que aparece en 1981, ampliando paulatinamente en los siguientes años su precisión.

Por último, es necesario comentar el auge de los datos desde un punto de vista social respecto al uso de la información geográfica, donde se empieza a entender la importancia de una buena gestión de los datos espaciales. En España han

proliferado algunos SIG que brindan servicios públicos a la ciudadanía permitiendo realizar un análisis de los datos espaciales a través de consultas a una base de conocimiento geográfica, de manera muy accesible para personas que no conocen las tecnologías, Esto hace necesaria la creación de nuevas vías de comunicación de la información espacial que mejoren su gestión. Es por ello que surgen las denominadas Infraestructuras de Datos Espaciales (*IDE*) (Masser, 2005), como por ejemplo el IDE Nacional de Estados Unidos (*NSDI*), que permiten gestionar y organizar las informaciones geográficas nacionales (Masser, 2005). Otro ejemplo es el sistema SIGPAC, un SIG que permite identificar geográficamente las parcelas declaradas, muy utilizado por el sector primario español (Almoguera Millán, Tarquis Alfonso, & Hernández Díaz-Ambrona, 2009).

Desde el punto de vista de las técnicas y teorías que fueron las bases de los SIG, éstas tuvieron grandes problemas en sus inicios debido al gran desconocimiento teórico de la verdadera función de un SIG y la falta de herramientas de creación de estos sistemas. En estos momentos se dejó muy de lado el análisis de los datos que por aquel entonces, era el fin con que el nacían. Simultáneamente, la cartografía cuantitativa sufrió un gran avance, por lo que se empiezan a desarrollar nuevas formulaciones que se unen al avance de los SIG de la época. Este avance estaba movido por las necesidades técnicas del momento y no por los fundamentos teóricos de algunos estudiosos (McHarg & Mumford, 1969) sobre los principios de las superposiciones de elementos, los cuales fueron introducidos posteriormente y forman parte esencial de cualquier SIG. Sin embargo, ya mucho antes se realizaban análisis manuales sobre cartografía, como el caso de *John Snow*, el cual utilizó mapas de puntos para identificar un brote de cólera en Inglaterra en la época de 1835 (Snow, 1855).

Una vez el análisis de los SIG va haciendo acto de presencia, se unen otras ramas científicas como la cartografía, a través de la utilización del estudio del relieve, y la estadística, para la ejecución de estadísticas espaciales. Sin embargo, lo que propició el gran avance de los Sistemas de Información Geográfica fue la evolución paralela de la informática, desde los años sesenta en adelante. Esto favoreció la evolución y diseño de los sistemas dando a estos la accesibilidad y facilidad de uso con la que cuentan hoy en día.

2.3.2 Herramientas para el desarrollo de los SIG

Actualmente existen varias soluciones disponibles en el mercado para el desarrollo de Sistemas de Información Geográfica, entre los que hemos destacado los siguientes para un estudio comparativo que nos permita analizar las virtudes y carencias de cada uno de ellos:

Tabla 1. Descripción de la plataforma CARTO




	Nombre	CARTO
	Compañía	Carto
	Lanzamiento	2011
	Descripción	Carto es un sistema SaaS (Software as a Service) que brinda herramientas para el análisis y visualización de datos espaciales en su plataforma web. (Paul Crickard, Eric van Rees, 2018)
	Características	La diferencia que brinda Carto con respecto a los demás sistemas es su facilidad de uso, su accesibilidad para usuarios sin conocimientos de programación, configurable y de fácil manejo gracias al diseño de su moderna interfaz.
	Licencias	Individual y Enterprise, ambas de pago.
	Documentación	Adecuada y pública.
	Página oficial	https://carto.com/

Tabla 2. Descripción de la plataforma MapBox.

	Nombre	MapBox
	Compañía	Mapbox
	Lanzamiento	2010
	Descripción	MapBox es un proveedor de imágenes geoespaciales de diferentes fuentes, tanto


		públicas como privadas, que brinda mapas personalizables y adaptados al usuario.
	Características	Lo que destaca de MapBox es la personalización de cada uno de los elementos de sus mapas, así como colores, plantillas, organización y un largo etcétera. También destacan sus SDK's existentes para todas las plataformas disponibles: iOS, Android, Web, Qt, Unity y MacOS.
	Licencias	Depende de los servicios contratados. Pago por uso.
	Documentación	Adecuada y pública. Destaca la publicación de muchos códigos de ejemplo en la plataforma GitHub.
	Página oficial	https://www.mapbox.com/

Tabla 3. Descripción del sistema MapInfo.

	Nombre	MapInfo
	Compañía	Pitney Bowes Software
	Lanzamiento	1986
	Descripción	MapInfo, en su última versión Pro, es una aplicación GIS de escritorio (desktop), dedicada al análisis de datos espaciales y su representación en mapas de localización para obtener información extraída de las visualizaciones y la relación en los datos. (Daniel, Loree, & Whitener, 2002)
	Características	Una de sus peculiaridades es el fácil tratamiento de los datos con el que cuenta, a través de la adición, edición y actualización de los conjuntos de datos de

		trabajo. Otras de sus características más importantes es el conjunto de herramientas de tratamiento de los datos que ofrece, el modelado de sus datos en diferentes clases para su manejo y su módulo estadístico de análisis.
	Licencias	Compra del software y licencia de desarrollador.
	Documentación	Completa y pública.
	Página oficial	https://www.pitneybowes.com/us/location-intelligence/geographic-information-systems/mapinfo-pro.html

Tabla 4. Descripción del sistema SmallWorld.

	Nombre	SmallWorld
	Compañía	GE Energy Connections
	Lanzamiento	1986
	Descripción	SmallWorld es un sistema de gestión de infraestructuras geoespaciales dedicadas a todo tipo de recursos: agua, gas, redes de gestión eléctrica o telecomunicaciones. (Kuang & Xiao, 2006)
	Características	La arquitectura permite su uso como gestor de datos geográficos y espaciales de una compañía, sin importar el tamaño de ésta. Además, el sistema se integra con muchas otras tecnologías con las que una organización gestiona sus recursos de infraestructuras de distribución, clientes u ordenes de trabajo entre otros.
	Licencias	Compra del producto software.
	Documentación	Inaccesible de forma pública.

	Página oficial	https://www.grintec.com/en/Smallworld
--	-----------------------	---

Sin embargo, y debido a que todos ellos son sistemas de pago, se decide utilizar el conjunto de software con el que trabaja y cuenta con licencia la compañía Eptisa, organización donde se han llevado a cabo los trabajos. Se trata de ArcGIS (Santiago, 2005), un Sistema de Información Geográfica desarrollado por la compañía ESRI, fundada en 1969.

ArcGIS es el conjunto de programas y soluciones GIS para la recopilación, organización, administración y análisis de la información geográfica (Santiago, 2005). ArcGIS está integrado por diferentes herramientas como son ArcMap, ArcCatalog, ArcToolbox, ArcScene y ArcGlobe. Su última versión integra todas ellas en una única plataforma llamada ArcGisPRO. Además, cuenta con otras herramientas que completan su catálogo como son ArcGIS Online, plataforma en línea de gestión de datos geográficos, ArcGIS Server y ArcGIS Mobile.

Entre las posibilidades que brindan las herramientas de ArcGIS se encuentran crear, compartir y utilizar visualizaciones geográficas de los datos espaciales en mapas, trabajar con información geográfica, gestionar datos espaciales y geográficos en bases de datos especializadas, realizar análisis espaciales de los datos y diseñar y crear aplicaciones apoyadas en el uso de mapas.

2.4 Introducción a Machine Learning

Vivimos una época nunca antes vista en el ámbito del almacenamiento y tratamiento de los datos. La cantidad de datos que manejamos hoy en día es de tal magnitud que se hace complicado realizar un análisis de todos ellos para obtener información relevante. Esto es debido a la disminución de los precios de los sistemas de información y de los sensores con los que captamos esos datos. Según una infografía hecha por una experta en análisis de datos, cada minuto, sí, cada minuto, se realizan 1 millón de accesos a la red social *Facebook*, se envían más de 41,6 millones de mensajes en *WhatsApp*, se mandan más de 188 millones de correos electrónicos, se ven 694,444 horas de vídeo en *Netflix*, y gastan en compras online más de 996,956 dólares (Lori Lewis, 2019).

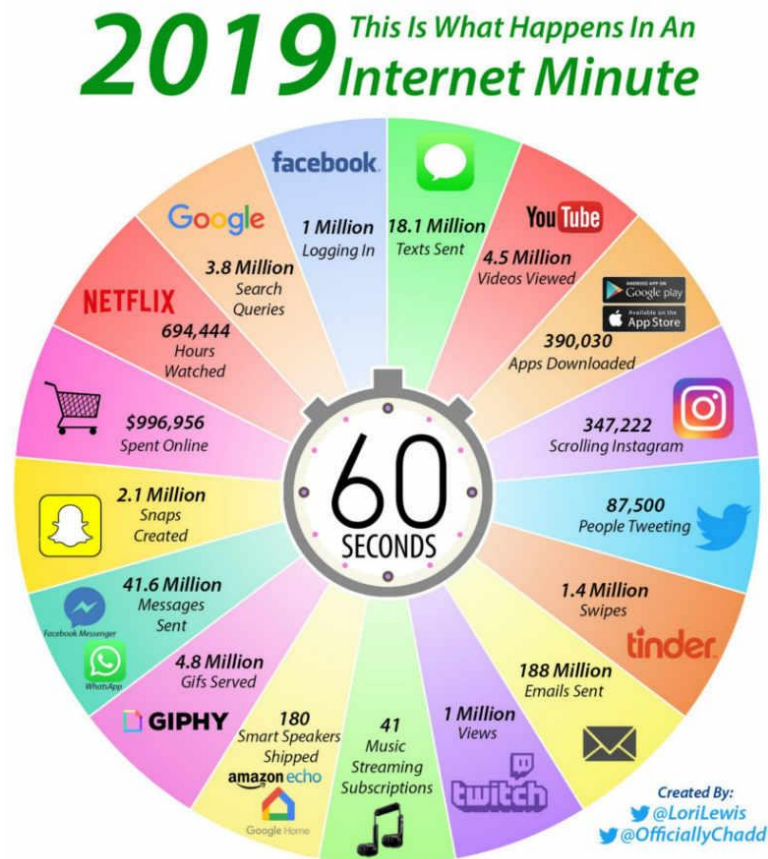


Figura 3. Infografía de los datos producidos por un usuario cada minuto.

Este incremento de la producción de datos es el alimento perfecto para el auge que están teniendo las ciencias de datos, especialmente el *Machine Learning*, “Aprendizaje Automático” en español. No es fácil definir el término *Machine Learning*, debido al cambio que sufre esta ciencia a diario, por lo que nos remitimos a uno de los expertos de la materia, Dan Fagella, CEO y fundador de Emerj, para obtener una definición válida en la que apoyarnos para explicar el término. Fagella definió el aprendizaje automático (Fagella, 2019) como “una ciencia que permite que las computadoras aprendan y actúen como lo hacen los humanos, mejorando su aprendizaje a lo largo del tiempo de una forma autónoma, alimentándolas con datos e información en forma de observaciones e interacciones con el mundo real”. Machine Learning se ha convertido en la ciencia que nos permite obtener conocimiento a partir de una información extraída de los datos, en la mayoría de los casos no visible a simple vista. Algunos usos de esta tecnología son:

- Asistentes virtuales: Algunos ejemplos son los más que conocidos asistentes del Smartphone de Apple Siri, el altavoz inteligente de Google Home, y la famosa Alexa de la compañía Amazon en sus Amazon Echo (Medina, Cabeza, & Peña, 2013).
- Servicios en redes sociales: Reconocimiento facial, identificación de imágenes (Díaz, 2013).
- Control de spam y filtros de software malintencionado (Awad & ELseuofi, 2011).
- Recomendación de productos: Ampliamente usado por las compañías para mostrarnos anuncios de acuerdo a nuestras últimas búsquedas (Perlich, Dalessandro, Raeder, Stitelman, & Provost, 2014).
- Análisis de videos: Usado para detectar patrones en cámaras de vigilancia, control de aforo en eventos o control de la ciudadanía, entre otros (Portilla Jiménez, 2018).

2.4.1 Vocabulario básico

Los algoritmos de Machine Learning son alimentados por datos normalmente almacenados en bases de datos. Es por ello por lo que las estructuras de datos usadas generalmente para el tratamiento de los datos sean matrices y vectores. Por ello usamos el término *registro*, muestra, o dato para referirnos a las filas de las matrices en las que se almacenan nuestros datos. Las columnas de estas tablas son *atributos* (características) de cada una de las muestras anteriores. En la mayoría de los casos también puede aparecer un atributo llamado *etiqueta* u *objetivo*, que será el valor que pretendemos predecir.

Algunos algoritmos con los que trabaja el Machine Learning tienen la posibilidad de obtener por parámetros algunos valores que configuran su ejecución. Estos valores son llamados *hiperparámetros*.

Además, llamamos *entrenar* un modelo al proceso mediante el cual un algoritmo de Machine Learning específico aprende a partir de unos datos especialmente dedicados a este proceso.

2.4.2 Tipos de Aprendizaje Automático

Aunque existen muchos y muy diversos tipos de aprendizaje automático, nuestro fin es dar una aproximación a estos métodos, por lo que nos centramos en los tipos candidatos a ser implementados en nuestro sistema:

2.4.2.1 Aprendizaje supervisado

Llamamos *aprendizaje supervisado* (Cambronero & Moreno, 2006) cuando se entrena un modelo con unos datos que cuentan con atributos o características y en el que cada registro tiene una etiqueta conocida. Gracias a esto el algoritmo puede llegar a predecir a través de la combinación dicha variable clase u objetivo. Existen dos tipos de aprendizaje supervisado (Cambronero & Moreno, 2006):

- *Regresión*: La variable clase u objetivo es una variable objetivo continua. Para predecirla se cuenta con un conjunto de variables predictoras o explicativas. El algoritmo trata de encontrar una relación entre las variables y la clase para ofrecer nuevas predicciones ante nuevos casos a partir de ese modelo.
- *Clasificación*: En este caso el objetivo del algoritmo es clasificar los datos de forma que pueda predecir las clases categóricas a las que pertenecen los datos del conjunto de entrenamiento, y crear un modelo o grupos con los que predecir la clase objetivo de los nuevos datos.

2.4.2.2 Aprendizaje no supervisado

En el aprendizaje no supervisado, no contamos con un conjunto de datos etiquetados, por lo que el objetivo es agrupar dichos datos tratando de buscar las relaciones y mejorar los agrupamientos. Distinguimos dos tipos de aprendizaje no supervisado:

- *Clustering*: Lo que se busca con las técnicas de agrupamiento o clustering es agrupar los datos a través del análisis y búsqueda de colectivos similares en cuanto a sus características. Estos grupos pueden no ser fijos y evolucionar con cada una de las ejecuciones del algoritmo de entrenamiento.
- *Reducción de la dimensionalidad*: En algunos casos el conjunto de datos con el que contamos posee un gran número de características o variables, lo que llamamos gran dimensionalidad. En estos casos se busca establecer correlaciones entre las características que permitan reducir el tamaño y ruido

del conjunto, para trabajar en un espacio más pequeño que mantenga el valor de sus datos.

2.4.2.3 Deep Learning

El *Deep Learning* es una tecnología que trata de llevar a cabo procesos de aprendizaje automático usando redes neuronales artificiales que imiten las de un ser vivo (Lauzon, 2012). Está compuesta por varios niveles jerarquizados. En el primer nivel gestiona la información de forma simple pasándola al siguiente nivel para su procesamiento, así sucesivamente hasta que los datos llegan al último nivel y se brinda una predicción. Esta predicción es valorada y comparada para mejorar el sistema. Los nodos con los que trabaja, también llamados neuronas, están conectados entre sí como si fuese una tela de araña. Los nodos de entrada y salida, en el primer y último nivel, están conectados gracias a los nodos interiores.

Uno de los métodos de Deep Learning más importante se conoce como *Aprendizaje reforzado* (Sutton, Barto, & others, 1998). Se basa en la experimentación con un modelo que mejora su rendimiento gracias a un agente, apoyado en recompensas que indican el acierto de cada acción. El agente trata de maximizar la recompensa modificando su comportamiento.

2.4.3 Proceso de construcción de modelos

Existe una metodología común a la mayoría de los procesos de construcción de un modelo de *Machine Learning*. En ellos se distinguen las siguientes tareas (Kompore, Bratko, Steinman, & Džeroski, 1994):

- *Preprocesamiento*. Es una de las tareas más importante a la hora de crear un modelo. Los datos no suelen estar optimizados para su tratamiento, y suelen tener errores, ruido, o ser incorrectos. Por ello existe un proceso previo de limpieza y adecuación de la base de conocimiento para su correcto análisis. Es una de las tareas imprescindibles y que se debe realizar siempre. Se realizan también técnicas de estandarización y normalización para adecuar los datos a las necesidades del modelo. En otras ocasiones también es posible que se tenga que reducir la dimensionalidad debido a que existen redundancia en las variables, por estar estas relacionadas entre sí. También

se debe dividir el conjunto de datos en dos tipos llamados de *entrenamiento* y de *prueba*, aleatorizando los datos.

- *Entrenamiento y selección del modelo.* Es muy importante elegir de manera adecuada el algoritmo a utilizar. Existen gran cantidad de ellos para cada uno de los problemas que resuelve el aprendizaje automático o *Machine Learning*.

También es importante elegir las métricas adecuadas para valorar y medir el rendimiento de los modelos (Tiwari, 2017), como por ejemplo el Error Cuadrático Medio (MSE) que se computa como (Tiwari, 2017):

$$MSE = \frac{1}{N} \sum_{i=1}^N (f_i - y_i)^2$$

También son usadas otras como la precisión de la clasificación (Tiwari, 2017), *recall*, *F1-score* o el área bajo la curva de funcionamiento del receptor (AUC). Además, podemos usar técnicas que optimicen los parámetros de los algoritmos que usamos, para encontrar la configuración óptima que maximice el rendimiento de nuestro algoritmo. Para ello se usan procesos de optimización de los hiperparámetros.

- *Evaluación.* Una vez obtenido el modelo se debe usar los datos de prueba o *test* para ajustar su rendimiento ante la predicción de nuevos datos, y medir su exactitud usando diferentes métricas. Una vez ajustado, tendremos un modelo funcional para usar con otros datos.

2.5 Conclusiones

Tras la lectura de este capítulo de presentación de los principales conceptos y tecnologías del trabajo, se conoce más de cerca el concepto de SIG, así como sus modelos y características más importantes. Esto ofrece una idea de todo el potencial que tienen estos sistemas, así como el contexto social y evolutivo en el que se encuentran. Además, se introducen los principales Sistema de Información Geográfica de la actualidad, definiendo el que se considera un paso por delante del resto, el sistema ArcGIS de la compañía ESRI.

A continuación, se ha presentado también una introducción al Machine Learning, definiendo la metodología de construcción de los modelos y los tipos de estos necesarios en capítulos posteriores.

Capítulo 3

Contexto y Metodología

3.1 Introducción

En este capítulo se da un contexto empresarial a la memoria, presentando las compañías participantes en este trabajo, ubicando geográficamente además las tareas llevadas a cabo, y ofreciendo una presentación de la metodología puesta en práctica durante el transcurso de los trabajos. A continuación, se hace un resumen temporal de las tareas desarrolladas, con el fin de planificar el tiempo y los recursos necesarios para el buen funcionamiento del proyecto.

3.2 Contexto del TFG

Este trabajo fin de grado ha sido realizado en una de las sedes con las que cuenta la compañía Eptisa. En concreto la sede ubicada en el Parque Científico y Tecnológico de Castilla-La Mancha, en la ciudad de Albacete, lugar donde realicé mis prácticas de grado y que ahora constituye mi lugar de trabajo habitual. Eptisa se define como (Eptisa, 2019) “una compañía multinacional de ingeniería, consultoría, tecnologías de la información y desarrollo institucional, económico y social con más de 60 años de experiencia. La calidad de sus servicios la convierte en referencia en los sectores de Transporte, Agua, Medio Ambiente, Industria, Energía, Edificación y Desarrollo Institucional Económico y Social. En relación a las actividades de las Tecnologías de la Información, Eptisa constituyó en el año 1991 la empresa Eptisa Tecnologías de la Información S.A. que en la actualidad tienen el liderazgo en Sistemas de Información Geográfica, así como

en servicios de consultoría tecnológica. Con presencia en más de 45 países y una red de 25 oficinas locales consolidadas, Eptisa es una empresa global que actúa en local, cercana al cliente y a la sociedad civil en la que opera”.

El proyecto desarrollado se ha desplegado en la ciudad peruana de Chimbote, capital de la provincia del Santa. Es ésta una de las veinte provincias que integran el departamento de Áncash, que a su vez es uno de los veinticuatro departamentos que componen la República de Perú. Su población actual se encuentra en torno a los 450.000 habitantes. Se caracteriza por ser una de las ciudades más importantes de Perú en cuanto al mercado de la pesca se refiere, contando con la denominación de *Capital del Trabajo* debido al centenar de fábricas que se ubican en la ciudad y que dan trabajo a muchos de sus habitantes. Por ello Chimbote es considerado uno de los mayores puertos pesqueros del país.

Este proyecto forma la parte de sistema de información geográfica a cumplir en un contrato único en el que participan dos empresas. Una de ellas con experiencia en la implantación de Sistemas de Información Geográfica. Otra con experiencia en la gestión integral del agua potable y saneamiento, la cual es responsable de la parte del catastro técnico y comercial del contrato.

La realización del proyecto se lleva a cabo por un equipo de 5 personas, repartidos geográficamente tanto en las diferentes oficinas fijas con las que cuenta Eptisa, en Albacete (España) y Lima (Perú), como la oficina de proyecto ubicada en Chimbote (Perú). Entre los integrantes del proyecto, destaca la diferenciación de roles en cuanto a las tareas encomendadas a cada trabajador, existiendo dos equipos de trabajo dentro del proyecto, uno dedicado a labores cartográficas y otro segundo equipo dedicado a tareas de programación, todo dirigido por un jefe de proyecto responsable y organizador de las diferentes funciones. Todas y cada una de las tareas que se realizan en este trabajo fin de grado fueron desarrolladas íntegramente por el equipo de programación, formado únicamente por el autor del mismo, bajo la supervisión del jefe de proyecto.

3.3 Fases del proyecto

En esta sección se define las fases realizadas para el desarrollo del presente TFG, desde su puesta en marcha y análisis previo, hasta su cierre, con la realización de la memoria y valoración de los resultados. Distinguimos las siguientes tres fases:

- **Fase de investigación e inicio.** En esta primera fase se estudiaron las tecnologías que serían usadas para la realización de la solución. Además, se realiza un examen crítico y valoración de las diferentes alternativas.
- **Fase de implementación de la solución.** Una vez terminada la fase anterior, se desarrolla el sistema, cuya tarea es a su vez dividida en diferentes etapas ligadas a los diferentes objetivos:
 - **Implementación de los scripts de codificación.** Es la primera tarea que se lleva a cabo y pretende ser un punto de partida preparatorio para la cartografía base con el fin de usarla en otros desarrollos ligados al proyecto.
 - **Implementación del sistema de clasificación de elementos.** Se modifica la planificación para incluir este objetivo dentro del proyecto cuando se detecta su necesidad.
- **Fase de creación de la memoria.** Una vez iniciado el proceso de implementación de la solución, se realiza de forma simultánea el proceso de documentación y preparación de esta memoria, esto facilita en gran medida plasmar los detalles del proceso.

En la siguiente tabla se describen las diferentes tareas realizadas, así como su planificación temporal y fases a las que pertenecen. En la Figura 4 se presenta el diagrama de Gantt de la planificación.

Tabla 5. Planificación del tiempo del proyecto en etapas de desarrollo.

Tareas	Fecha Inicio	Fecha Fin	Días	Fases
Investigación y formación	01/08/2019	06/08/2019	5	Fase de inicio
Estudio de tecnologías	07/08/2019	08/08/2019	1	Fase de inicio
Formación específica	09/08/2019	14/08/2019	5	Fase de inicio
Implementación de los scripts de codificación	14/08/2019	18/09/2019	35	Fase de implementación

Implementación del sistema de clasificación	19/09/2019	15/10/2019	26	Fase de implementación
Elaboración de la memoria	25/08/2019	20/11/2019	87	Fase de documentación
Valoración de resultados y cierre de proyecto	21/11/2019	23/11/2019	2	Fase de documentación

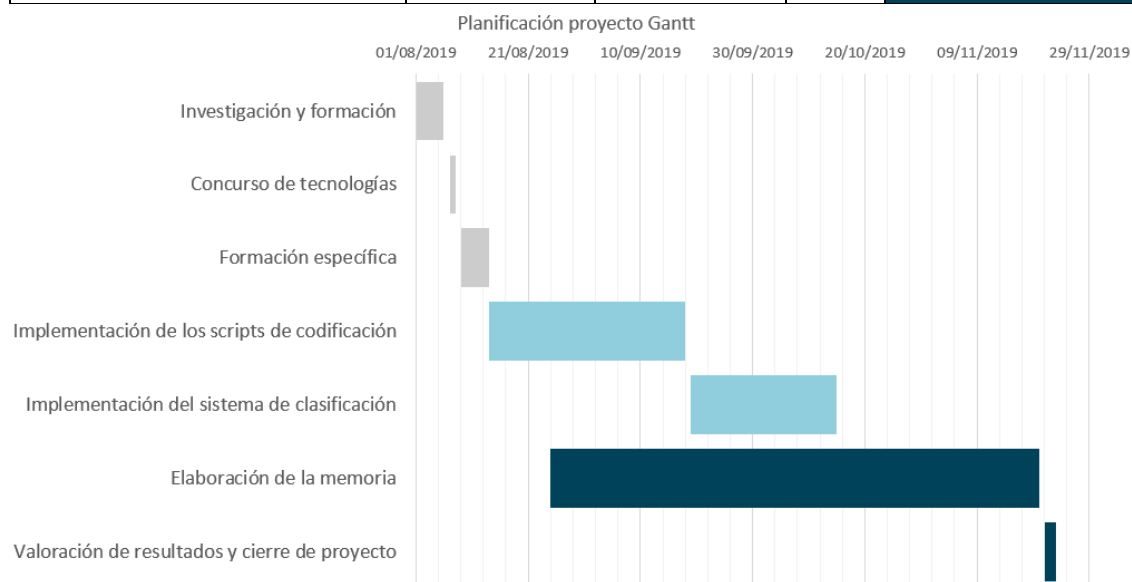


Figura 4. Diagrama de Gantt que muestra la planificación del tiempo del proyecto.

3.4 Conclusiones

Es este capítulo se conoce más de cerca la organización de las empresas participantes en el proyecto, en concreto, de la compañía Eptisa, en la que se realizan los trabajos documentados en esta memoria. Además, se desglosan y explican las diferentes fases en las que se divide el proyecto, obteniendo una estimación temporal de los trabajos a realizar.

Capítulo 4

Codificación de la cartografía base

4.1 Introducción

Tal y como se estableció en el Capítulo 1, uno de los subobjetivos de este proyecto es la codificación de una cartografía base, proporcionada por el cliente. Con ello se permite identificar los elementos cartográficos contenidos en ella de manera única e inequívoca. Para ello el cliente fija unos requisitos previos acerca de la metodología de codificación a seguir y la simbología o codificación de dicha numeración. Entre los elementos cartográficos a numerar encontramos subsectores, manzanas y parcelas catastrales (lotes). En este Capítulo, se describe cómo se ha automatizado dicha codificación como preámbulo necesario para el inicio de los trabajos en campo que se requieren para construir el catastro técnico y comercial que la empresa necesita.

En este capítulo también se hace un estudio de necesidades de las tecnologías a utilizar, barajando varias opciones y diciendo cuál de ellas se adapta mejor a la construcción de la solución.

4.2 Convenio de visualización de la cartografía base

Antes de mostrar cualquier cartografía en los siguientes capítulos, se hace necesario establecer una organización, en este caso por colores, de los diferentes elementos de ésta que permita su comprensión. Esto es posible gracias a las opciones de visualización que nos brindan los Sistemas de Información Geográfica.

Este convenio facilitará la identificación exacta y clara por el lector de todo elemento cartográfico mostrado en las imágenes de este trabajo, lo que mejorará la atención y comodidad durante la lectura de este texto. Así, se establece la siguiente clasificación:

Tabla 6. Convenio de colores para la visualización de la cartografía

CAPA	COLOR	CÓDIGO <i>RGB</i>
Sectores		<i>rgb</i> (195,247,198)
Subsectores		<i>rgb</i> (255,255,190)
Manzanas		<i>rgb</i> (255,190,190)
Lotes		<i>rgb</i> (190,232,255)
Zona Piloto		<i>rgb</i> (255,211,127)
Pueblos		<i>rgb</i> (212,199,252)
Esquinas		<i>rgb</i> (76,230,0)

Se añaden códigos *RGB*, codificación *red*, *green*, *blue*, de los colores para su mejor identificación. Estos códigos son además utilizados para la configuración del entono necesaria para la captura imágenes que se han incluido en este trabajo.

4.3 Cartografía base

La primera tarea que se requiere es codificar la cartografía base con la que cuenta el cliente para poder identificar cada uno de los elementos que pueden ser empleados durante la toma de datos en campo. Para ello dotamos a los elementos de una numeración única que permita identificarlos de manera unívoca. Es con tal fin que se desea crear unos procesos que se ejecuten de manera automática y que logren numerar los elementos de acuerdo con los requisitos previamente expuestos.

La cartografía base con la que se empieza a trabajar es una geodatabase exportada de la base de datos del proyecto alojada en un servidor *SQL Server*, lo cual hace más cómoda la tarea de testeo y vuelta atrás ante cualquier fallo.

Antes de seguir definiendo la cartografía, se hace oportuno explicar el término geodatabase y sus características para tener una visión más amplia del contexto en el que nos encontramos. Se define *geodatabase* (ESRI, 2019a) como “una colección de datasets geográficos de varios tipos contenida en una carpeta de sistema de archivos común, una base de datos de Microsoft Access o una base de datos relacional multiusuario *DBMS* (por ejemplo, *Oracle*, *Microsoft SQL Server*, *PostgreSQL*, *Informix* o *IBM DB2*). Las geodatabases tienen diversos tamaños, distinto número de usuarios, pueden ir desde pequeñas bases de datos de un solo usuario generadas en archivos hasta geodatabases de grupos de trabajo más grandes, departamentos o geodatabases corporativas a las que acceden muchos usuarios”. Por lo que consideramos las geodatabases como las estructuras de almacenamiento para permitir el tratamiento de datos del Sistema de Información Geográfica. Concretamente, mediante el sistema ArcGIS, creado por la compañía ESRI, el cual es uno de los sistemas de análisis y gestión de datos geográficos más importantes en el mundo del SIG. Además cuenta con la comunidad de desarrolladores e investigadores más grande del mundo (Olmedo, 2006).

La geodatabase inicial cuenta con varias capas o *layers* las cuales contienen las diferentes geometrías de los elementos a numerar y que forman la cartografía, que son:

Tabla 7. Definición de la capa Sectores.

NOMBRE	DESCRIPCIÓN	NÚMERO DE ELEMENTOS	REFERENCIA ESPACIAL
Sectores	Sectores operativos de la cartografía.	10	UTM_Zone_17_Southern_Hemisphere

PREVISUALIZACIÓN

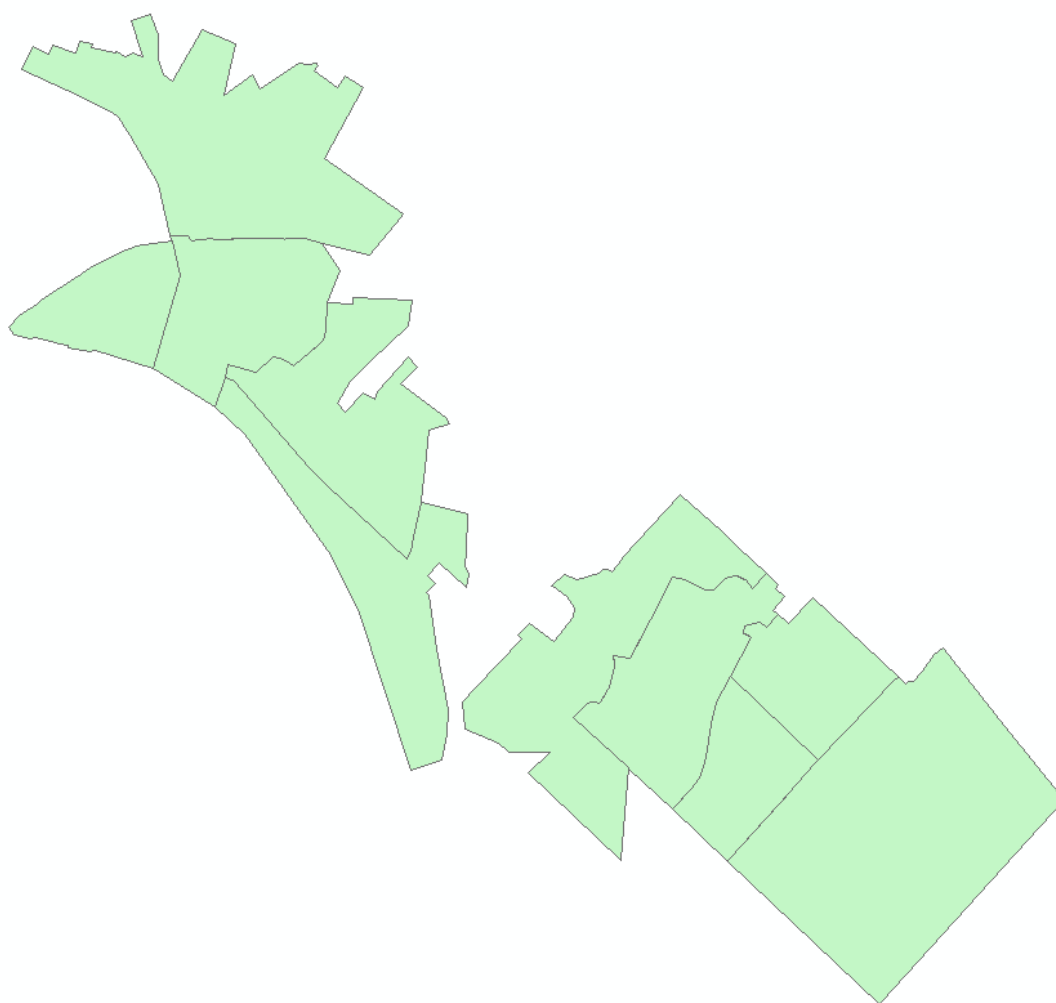


Figura 5. Imagen previa de la capa Sectores.

Tabla 8. Definición de la capa Subsectores.

NOMBRE	DESCRIPCIÓN	NÚMERO DE ELEMENTOS	REFERENCIA ESPACIAL
Subsectores	Subdivisión de la capa sectores.	30	WGS_1984_UT M_Zone_17S

PREVISUALIZACIÓN:

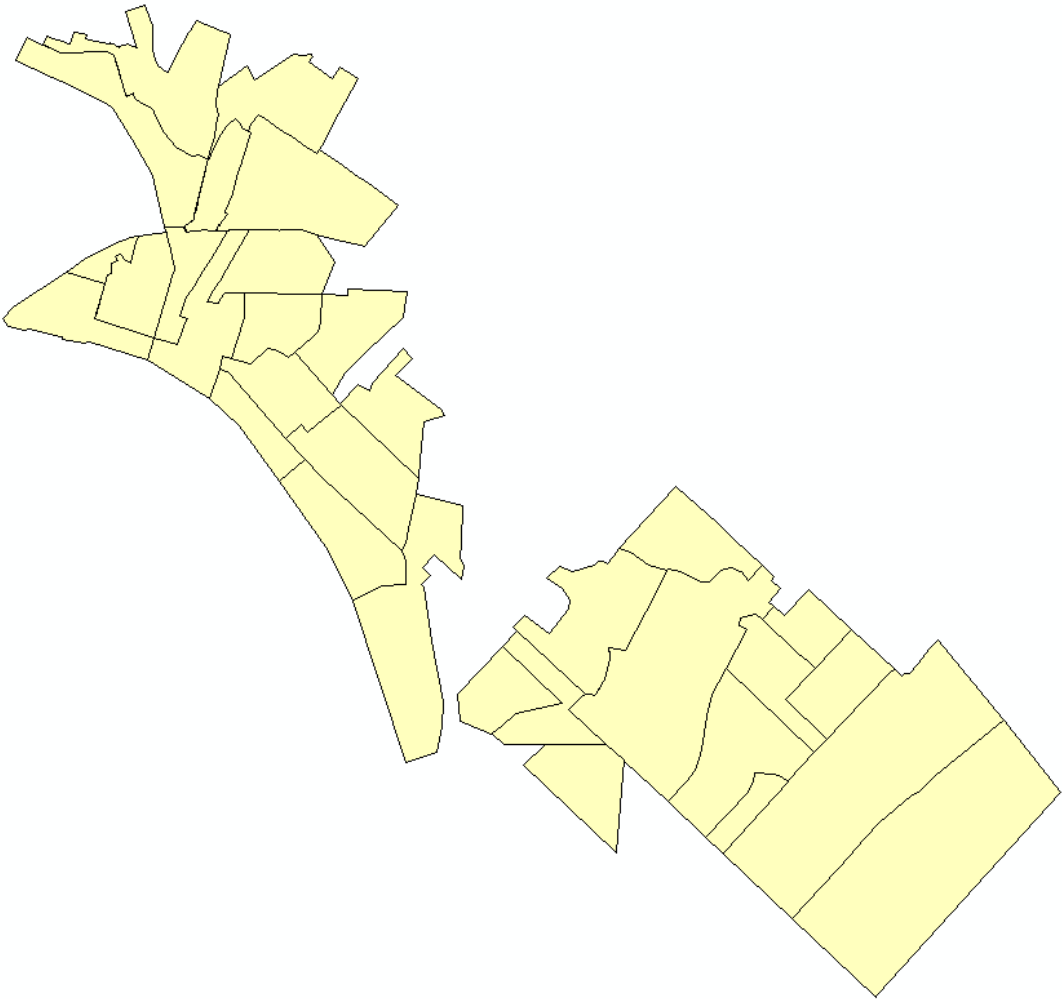


Figura 6. Imagen previa de la capa Subsectores.

Tabla 9. Definición de la capa Manzanas.

NOMBRE	DESCRIPCIÓN	NÚMERO DE ELEMENTOS	REFERENCIA ESPACIAL
Manzanas	División de espacios urbanos edificados o con fines de edificación.	4170	WGS_1984_UT M_Zone_17S

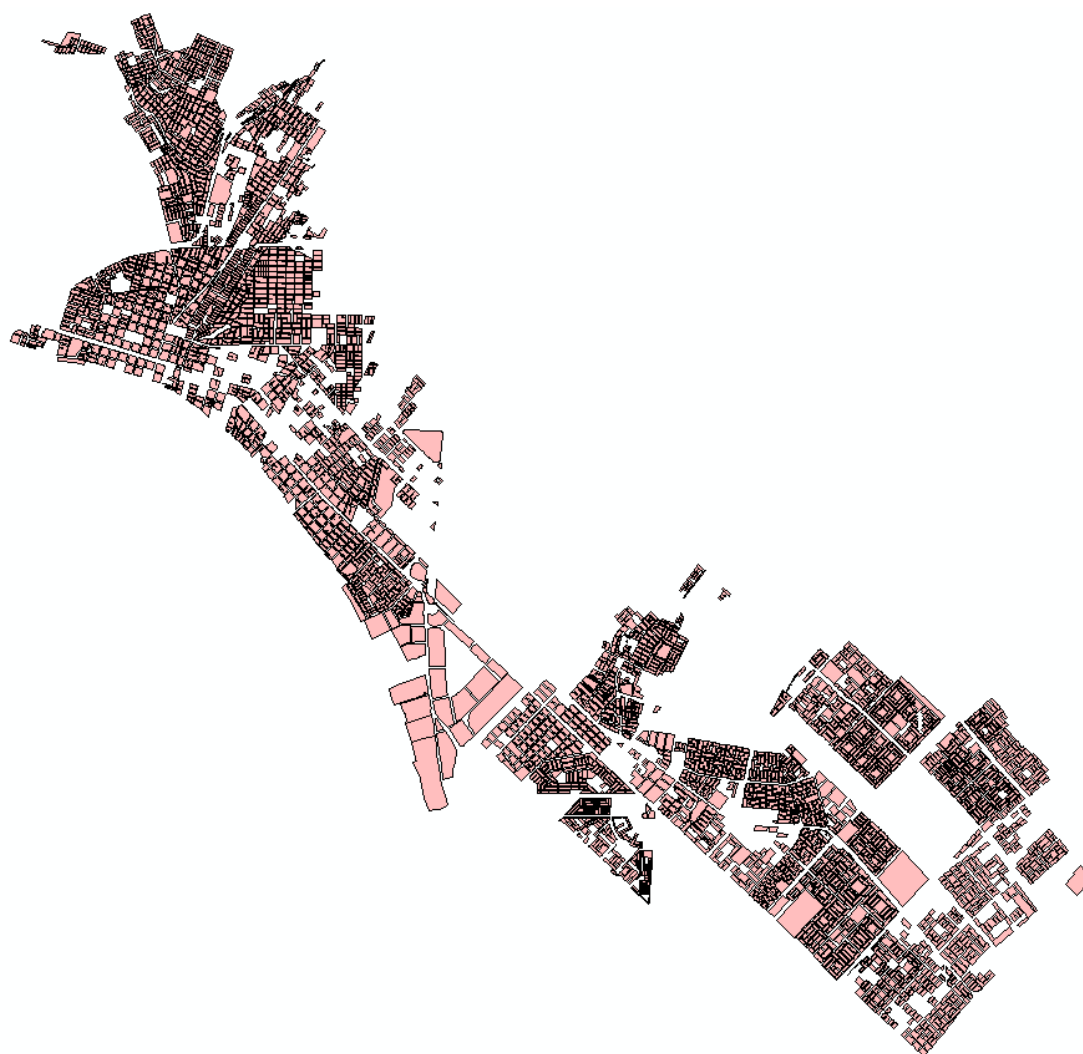
PREVISUALIZACIÓN:

Figura 7. Imagen previa de la capa Manzanas.

Tabla 10. Definición de la capa Lotes.

NOMBRE	DESCRIPCIÓN	NÚMERO DE ELEMENTOS	REFERENCIA ESPACIAL
Lotes	Subdivisión de la capa anterior en parcelas catastrales.	98999	WGS_1984_UTM_Zone_17S

PREVISUALIZACIÓN



Figura 8. Imagen previa de la capa Lotes.

Tabla 11. Definición de la capa Zona Piloto.

NOMBRE	DESCRIPCIÓN	NÚMERO DE ELEMENTOS	REFERENCIA ESPACIAL
Zona Piloto	Área de elementos cartográficos delimitada para testear el proceso de catastro.	1	WGS_1984_UTM_Zone_17S

PREVISUALIZACIÓN

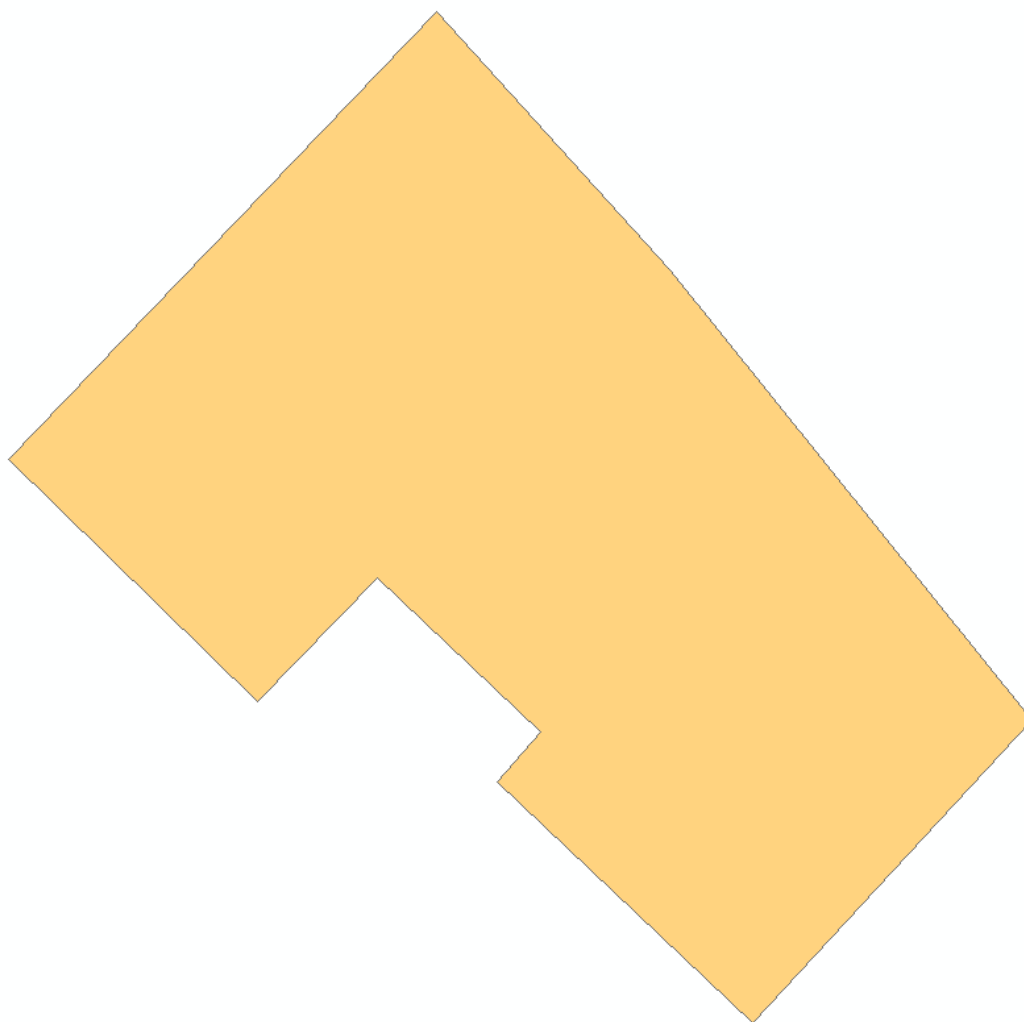


Figura 9. Imagen previa de la capa Zona Piloto.

Tabla 12. Definición de la capa Pueblos.

NOMBRE	DESCRIPCIÓN	NÚMERO DE ELEMENTOS	REFERENCIA ESPACIAL
Pueblos	División de regiones de Chimbote y Nuevo Chimbote	118	WGS_1984_U TM_Zone_17S

PREVISUALIZACIÓN

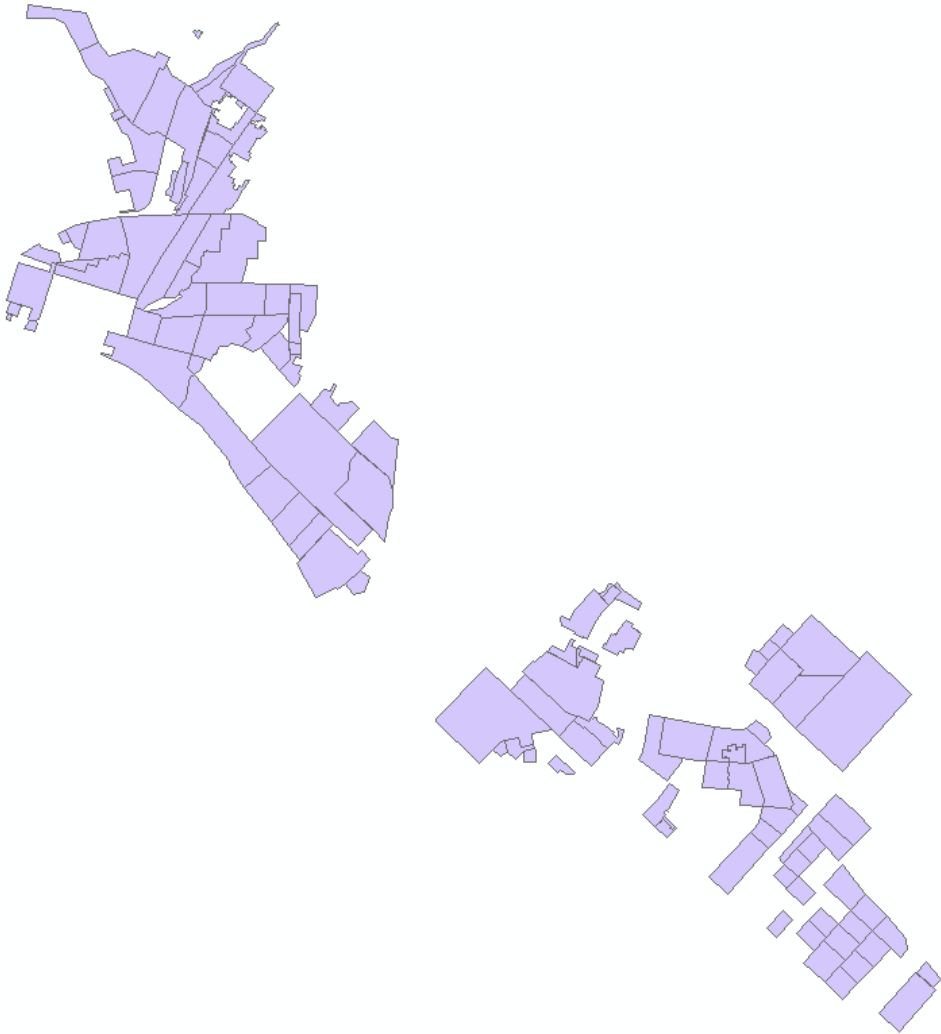



Figura 10. Imagen previa de la capa Pueblos.

Tabla 13. Definición de la cartografía base.

CARTOGRAFÍA BASE
PREVISUALIZACIÓN
 <p data-bbox="478 1534 1056 1572">Figura 11. Imagen previa de la cartografía base.</p>

4.4 Estudio de la tecnología

En esta sección se hace un breve estudio de la problemática a resolver y las posibles soluciones con las que abordarla. Para ello consideramos dos elementos clave de estudio, que son:

- Lenguaje de programación usado.
- Paquete de análisis de datos geográficos usado.


Tal y como se ha comentado, el primero de ellos es la elección del lenguaje de programación óptimo para el tratamiento de los datos y su procesamiento. Hoy en día existen infinidad de lenguajes, cada uno con gran cantidad de posibilidades ante todo tipo de retos o problemas a resolver. Sin embargo, en esta ocasión y dadas las características del problema a abordar, como son el tratamiento masivo de datos de carácter espacial, requiriendo esto de unas estructuras de datos que permitan la gestión eficiente de ellos y estén preparadas para su manejo, se decide trabajar con el lenguaje *Python*.

Python es un lenguaje de alto nivel, multiplataforma, de propósito-general, interpretado y orientado a objetos. Su uso está extendido para tareas de Inteligencia Artificial, Big Data, Data Science y diferentes frameworks de desarrollo web como Django. Según se puede leer en (Molina Ríos, Loja Mora, Zea Ordóñez, & Loaiza Sojos, 2016), entre sus principales ventajas se encuentran:

- Simplicidad y elegancia en el código.
- Soporta objetos y estructuras de datos de alto nivel: strings, listas, diccionarios, etcétera.
- Soporta múltiples formas de organización del código: funciones, módulos, clases y paquetes.
- Alta velocidad en el desarrollo y gran rendimiento.
- No requiere compilación.
- Open Source.
- Se puede utilizar en varias plataformas.

Además, es uno de los lenguajes que cuenta con una API disponible (Toms, 2015) para desarrolladores en el sistema ArcGIS usado. Este programa de desarrolladores incluye entre otros soporte para los siguientes lenguajes de programación:

Tabla 14. Lenguajes de programación integrados por ArcGIS para desarrolladores.

	<i>ArcGIS API for JavaScript</i>
	<i>ArcGIS Runtime SDK for Android</i>
	<i>ArcGIS Runtime SDK for iOS</i>
	<i>ArcGIS Runtime SDK for Java</i>
	<i>ArcGIS Runtime SDK for .NET</i>
	<i>ArcGIS Runtime SDK for Qt</i>
	<i>ArcGIS API for Python</i>
	<i>ArcGIS Arcade</i>
	<i>ArcGIS REST API</i>

También y como segundo punto de estudio, *ArcGIS* cuenta con gran cantidad de paquetes de apoyo al acceso y manejo de datos para el lenguaje *Python*, entre ellos destacamos el paquete *ArcPy*.

Se puede obtener una definición más exacta en la documentación oficial de *ArcGIS* (“¿Qué es ArcPy?—Help | ArcGIS for Desktop,” n.d.), en la que se define *Arcpy* como “un paquete de sitio que se basa en el exitoso módulo *arcgisscripting* y lo sucede. Su objetivo es crear la piedra angular para una manera útil y productiva de realizar análisis de datos geográficos, conversión de datos, administración de datos y automatización de mapas con Python”.

Por tanto, se decide usar esta combinación de herramientas para el trabajo a realizar. Se trabajará bajo la creación de scripts escritos en Python, con el apoyo del paquete de *ArcPy* para todo lo relativo al manejo y tratamiento de los datos. Además, se hará uso de algunas funciones de la caja de herramientas de

Administración de datos. Estas nos permiten ejecutar funcionalidades del poderoso software de ArcMap en código, permitiendo facilitar muchas de las tareas que se llevarán a cabo en el contexto de este TFG.

4.5 Codificación de subsectores

Antes del inicio de las tareas de numeración de la cartografía, se debe realizar una numeración de los subsectores. Esta tarea ya viene completada en la cartografía base, dada por un atributo en la tabla subsectores llamado “*Subsector*”, que almacena el código de sector y subsector en el siguiente formato (SS-B).

Por ello, el único trabajo que se ha de realizar es la separación del valor en sus respectivos atributos “*CODSECTOR*” y “*CODSUBSECTOR*”, realizado en el script “*tecnico_codificar_subsectores.py*”. Esta primera organización de atributos no es un simple capricho de programador, se hace con la intención de facilitar el acceso posterior a los datos evitando continuas divisiones del atributo para su utilización.

4.6 Numeración de manzanas

El primer requisito a desarrollar como parte de la codificación es la asignación de códigos únicos de manzana. Estas deben tener un código que permita identificarlas de forma unívoca. Se adopta la siguiente codificación: SSBMMM

- S: 2 dígitos para la identificación del Sector
- B: 1 dígito para la identificación del Subsector
- M: 3 dígitos para la identificación de la Manzana

Según la metodología de codificación propuesta por las necesidades del cliente se establece el siguiente requisito:

- **[RF1.1]:** La numeración de las manzanas comenzará a partir del 001 con el polígono de la manzana que se encuentre ubicado más al norte y si hubiera varios, aquella ubicada en el extremo noroeste, continuando de forma correlativa de izquierda a derecha y de arriba hacia abajo en zigzag.

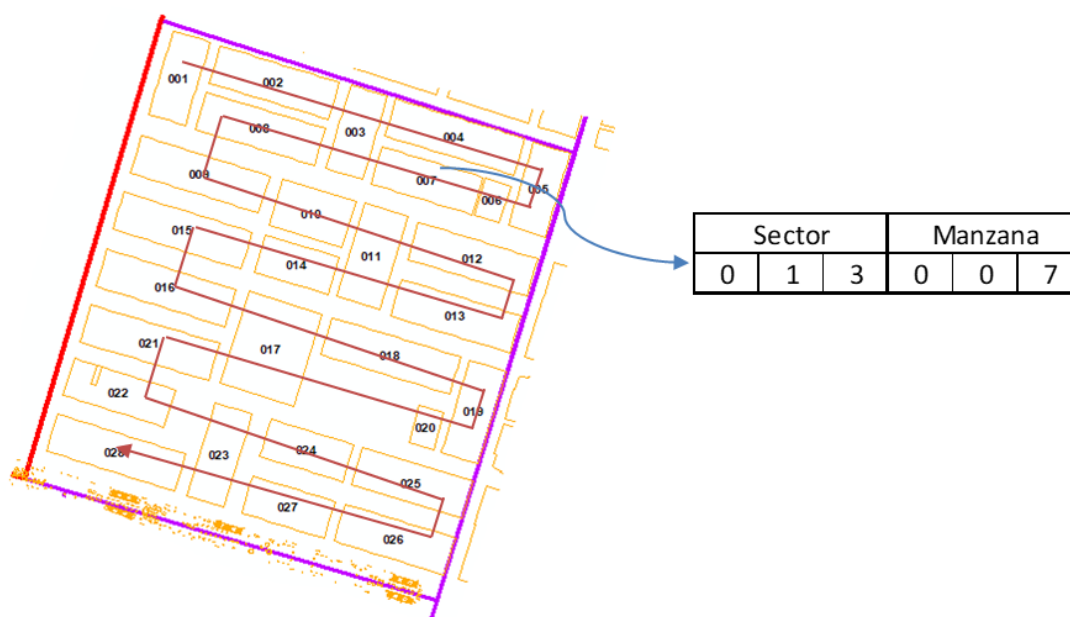


Figura 12. Requisitos de numeración de manzanas.

Previo a la numeración, se hace necesaria la creación de otro script que asigne previamente el código de sector y subsector en el que se encuentran las manzanas, de acuerdo con su posición, para su filtro posterior por dichos atributos.

4.6.1 Codificación de manzanas

Para ello se crea el script “*tecnico_codificar_manzanas.py*”, el cual, a través de uniones espaciales, logra asignar un código de sector y subsector a cada manzana de acuerdo con la superposición que ocupa con respecto a los espacios de sectores y subsectores.

El proceso que sigue el script es recorrer cada uno de los subsectores que existe en la clase de entidad “Subsectores” de la cartografía base. Para cada uno de esos subsectores ejecuta la función de análisis espacial de ArcGIS for Desktop “SelectLayerByLocation_management”. Concretamente se utiliza la función COMPLETELY_WITHIN (Completamente dentro) que permite determinar que un subsector está completamente contenido dentro de su sector padre, sino estaríamos ante un fallo en la cartografía tratada. Gracias a este script, el resultado que se obtiene es una capa de “Manzanas” clasificada por sector y subsector en sus nuevos campos “CODSECTOR” y “CODSUBSECTOR”, quedando excluidas y pendientes de revisión todas las manzanas que no

pertenecen a ningún subsector, ya que esto no es posible y se debe corregir en la definición de la cartografía.

4.6.2 Algoritmo de numeración de manzanas

Una vez queda clasificada la capa “Manzanas” de acuerdo con el sector y subsector al que pertenece cada uno de sus elementos, se define el algoritmo de numeración de las manzanas. Para ello se crea el script automático de análisis espacial, por nombre “*técnico_numerar_manzanas.py*”, el cual asigna números de manzana de acuerdo con los requisitos, asignando a cada una de ellas el código de sector y subsector correcto donde se sitúan.

Aunque no se consigue encontrar una solución ideal al problema, finalmente se llega a una aproximación que brinda muy buenos resultados de numeración. Los problemas para encontrar una solución ideal son debidos a que los subsectores de la cartografía no se encuentran situados en la misma posición geométrica, y a que la cantidad de casuísticas encontrada hace muy difícil encontrar una solución mejor. A todo esto, se le suma la vaga definición de los requisitos del sistema por parte del cliente. Se propone un algoritmo que numere según los criterios del objetivo RF1.1, los cuales marcan que la numeración de las manzanas comenzará a partir del identificador número 001 con el polígono de la manzana que se encuentre ubicado más al norte. Si hubiese varias, se comenzaría con aquella ubicada en el extremo noroeste, continuando en forma correlativa de izquierda a derecha y de arriba hacia abajo, similar a la sectorización, es decir, en zigzag.

Para ello el algoritmo busca la manzana más al norte y traza sobre el subsector, donde está contenida dicha manzana, una banda horizontal de tamaño igual a la altura de la manzana tratada, como podemos ver en la Figura 18.

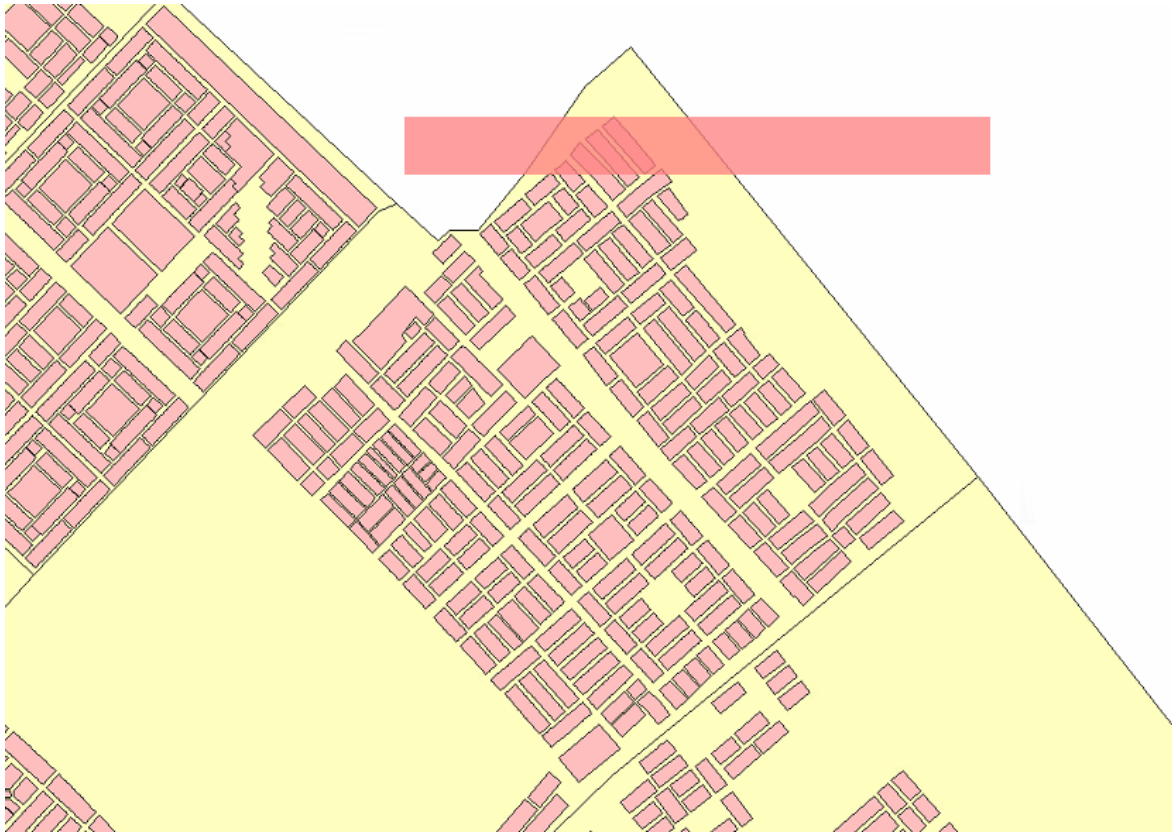


Figura 13. Banda horizontal de apoyo para la numeración de manzanas.

A continuación, se escogen candidatas para numerar todas las manzanas cuyo centroide o centro de gravedad se encuentre dentro de dicha banda. Esto nos permite suavizar la penalización que puede tener la orientación geométrica de una manzana con respecto a sus vecinas.

Lo siguiente que se ejecuta es el cálculo del punto más al noroeste de la banda. Se numeran las manzanas candidatas que están dentro de la banda horizontal de acuerdo con la distancia a dicho punto. De esta manera se obtiene una numeración horizontal perfecta de acuerdo con los requisitos. Sin embargo, se sacrifica la especificación de numerar primero la más al norte, ya que sería imposible seguir numerando las manzanas candidatas dentro de la banda horizontal que se encuentren más al oeste que ella. En la Figura 19 se puede ver cómo, tras la primera iteración del algoritmo, el estado de las manzanas candidatas deja de ser ideal, haciendo más complicado la continuación de su análisis. Por tanto, se podría definir el pseudocódigo del algoritmo de numeración como:

```
# Método para obtener la manzana candidata que se encuentra más al norte.
```

```

def get_manzana_mas_al_norte(lista_manzanas):
    for manzana in lista_manzanas:
        for esquina in manzana:
            if esquina.Y > north_block.Y:
                north_block = esquina;
    return north_block;

# Método que calcula las manzanas contenidas dentro de la banda.
def calcular_banda(lista_manzanas, manzana_norte, punto_norte):
    for manzana in lista_manzanas:
        if manzana_norte.YMin < manzana.Y < manzana_norte.YMax:
            manzana.distancia = distancia_punto_norte();
            banda.append(manzana);
    return banda;

def main():
    for subsector in subsectores_cartografía:
        envelop = subsector.extent;
        manzanas = get_lista_manzanas();
        # Mientras queden manzanas sin numerar
        # Iteramos sobre las manzanas del subsector
        while len(manzanas) > 0:
            # Se calcula la manzana más al norte
            manzana_norte = get_manzana_mas_al_norte()
            # Se crea banda horizontal y se calculan manzanas dentro de la banda
            banda_horizontal = calcular_banda(
                manzanas,
                manzana_norte,
                envelop.XMin if not utilities.is_even(band_num) else envelop.XMax
            )
            manzanas_candidatas = manzanas_dentro_banda(banda_horizontal)
            # Se numeran las manzanas de la banda
            ordenar_manzanas = ordenar_manzanas_por_distancia();
            for orden_manzana in ordenar_manzanas:
                resultado.append(orden_manzana);
            manzanas = manzanas - resultado;
    put_BBDD(resultado_manzanas);

```

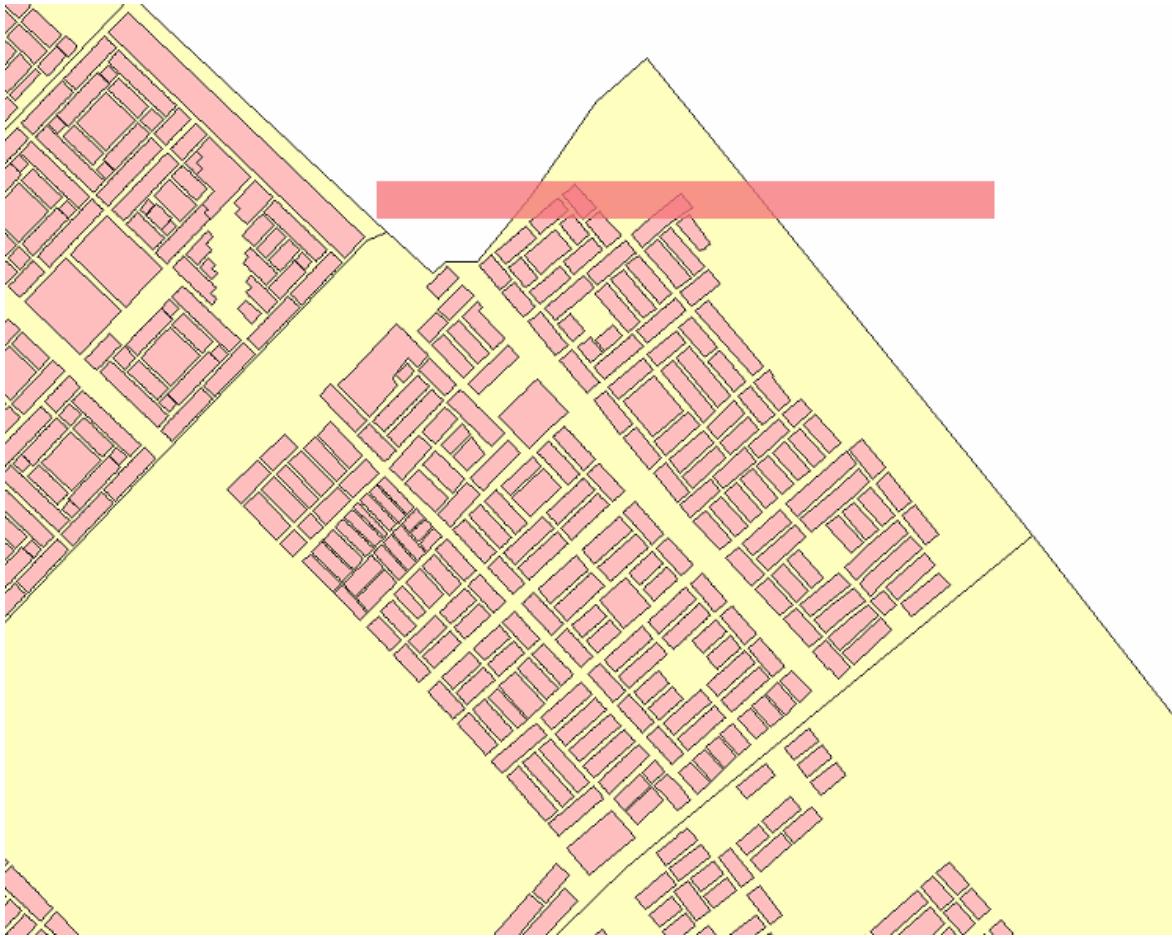


Figura 14. Segunda iteración del algoritmo de numeración de manzanas.

Se repite este proceso hasta que todas las manzanas del subsector queden numeradas, y se asegura por tanto la integridad del algoritmo al tener la certeza de que cada manzana estará contenida en al menos una banda horizontal.

Al completar todas las iteraciones, el resultado que se obtiene es algo parecido a lo que podemos ver en la Figura 20, un conjunto de bandas horizontales superpuestas o no, que contienen manzanas, las cuales se numeran en zigzag, cumpliendo por tanto los requisitos ordenados.

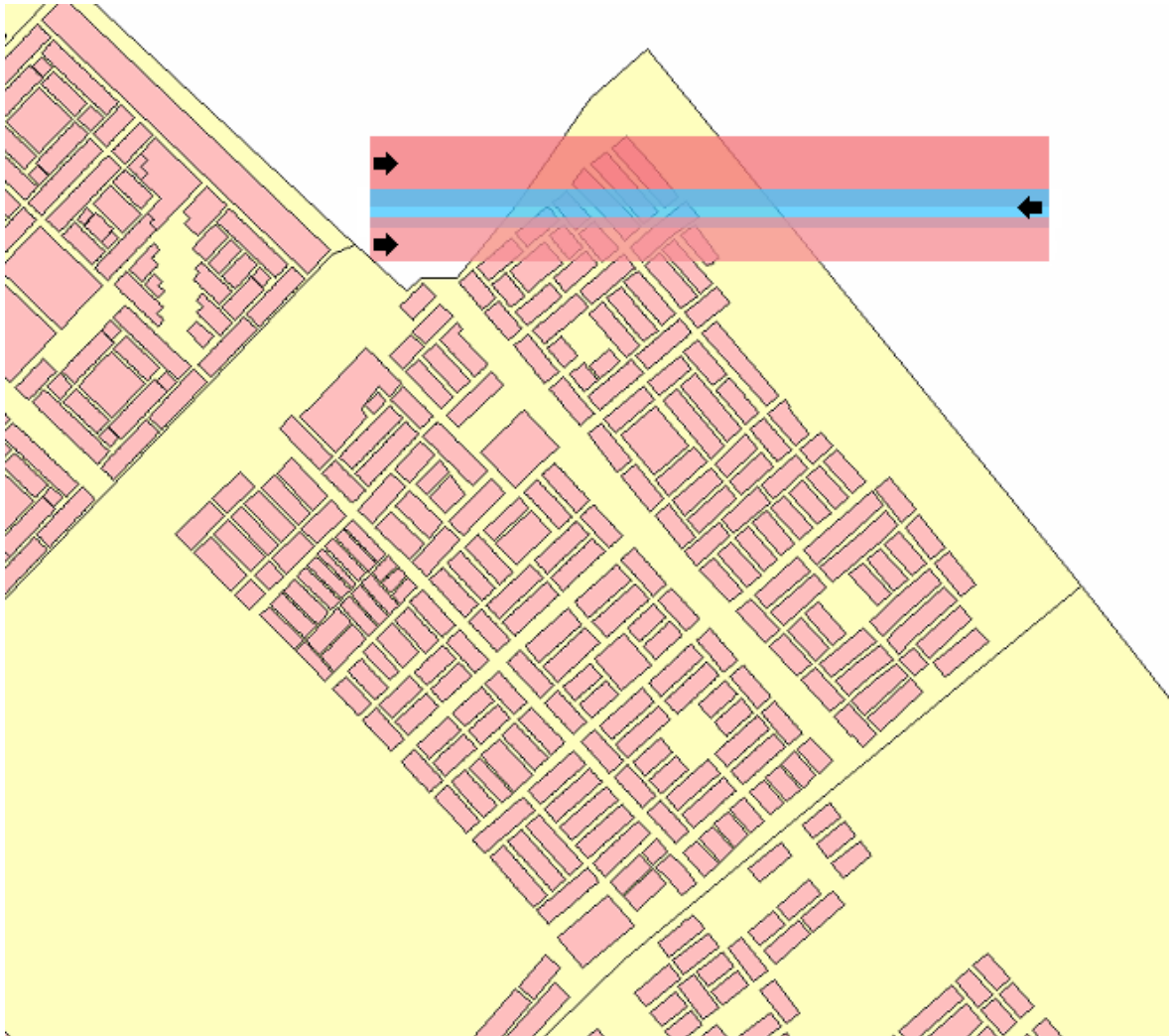


Figura 15. Representación visual del resultado del avance del algoritmo de numeración.

4.7 Numeración de esquinas

El siguiente requisito que se aborda es la numeración de esquinas:

- **[RF1.2].** Se han de numerar todas las esquinas de una manzana y generar su correspondiente capa de esquinas en el sistema GIS. La unidad de codificación es el esquinero, relacionado con las manzanas que lo conforman.

Dado que las manzanas tienen un código único, el valor del esquinero se construye a partir de la manzana y la esquina que se toma como referencia. Así, el código de esquina tiene el formato SSBMMME donde a la codificación de manzanas anteriormente descrita en la Sección 4.6.1 se le añade E: 2 dígitos para la identificación de la Esquina. Las esquinas se numeran, en cada manzana, comenzando en la esquina noroeste (NO), en sentido horario.

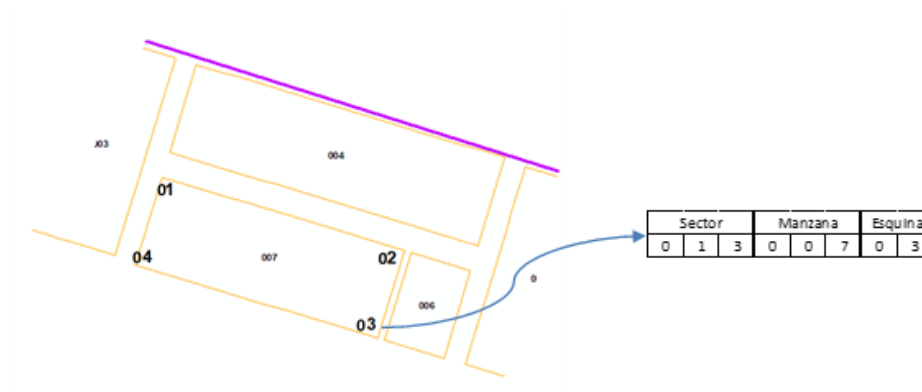


Figura 16. Requisitos de numeración de esquinas

Para ello se crea el script “*técnico_numerar_esquinas.py*”, descrito en la Sección 4.7.2, el cual es el encargado de recorrer todas las manzanas y numerar sus esquinas de acuerdo con los requisitos. Sin embargo, y previo a esta tarea, es necesario normalizar la cartografía tal y como se describe en la siguiente sección.

4.7.1 Normalización de elementos de la cartografía base

La cartografía base que se recibe por parte del cliente no suele ser perfecta. Muchas de las geometrías se encuentran mal ubicadas o formadas, por lo que se requiere de un trabajo paralelo de comprobación y modificación de la misma. El trabajo de corrección es difícil y tedioso, ya que no existe manera alguna de automatizar dicho proceso totalmente, por lo que la tarea debe ser llevada a cabo conforme se van descubriendo los fallos.

Es precisamente uno de estos problemas el que se encuentra a la hora de numerar las esquinas. Las geometrías de las manzanas no están trazadas correctamente y contienen vértices que no se corresponden con esquinas, como se ve en la Figura 22. Esto requiere de un proceso de normalización de las geometrías previo, que se ejecuta durante la ejecución del script de numeración de esquinas. En un primer paso nos apoyamos en la función *SimplifyPolygon* de la caja de herramientas Cartografía de ArcMap. Esta herramienta sirve para simplificar y refinar los datos espaciales de la cartografía con la que se trabaja, eliminando algunos de los vértices que definen la forma de las geometrías de acuerdo con los criterios configurados por parámetro.

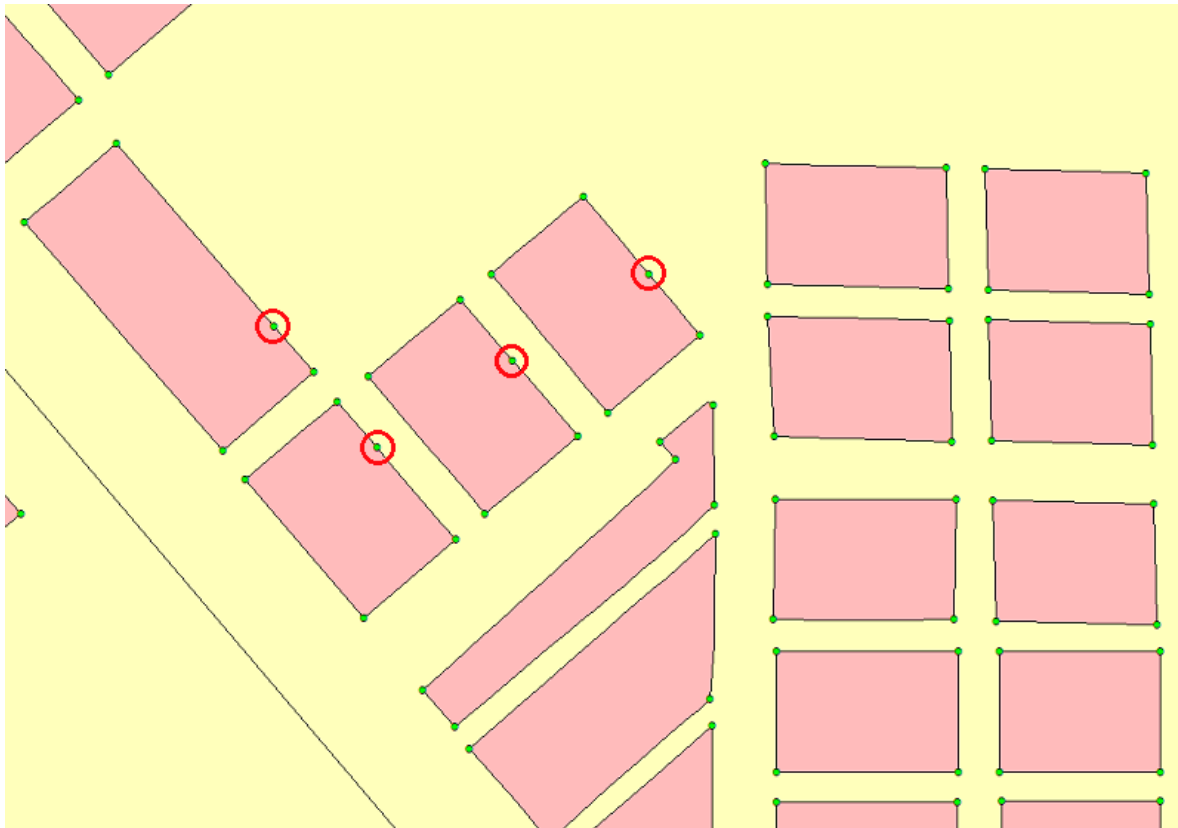


Figura 17. Vértices mal codificados en manzanas de la cartografía.

La normalización de las manzanas con un desplazamiento máximo permisible de 2 metros (especificado por parámetro) se realiza ejecutando un algoritmo de eliminación de puntos manteniendo sólo los llamados críticos. El resultado es una capa de manzanas mejor definida en sus límites, quedando sus vértices bien definidos espacialmente, y preparada para ser numerada.

4.7.2 Algoritmo de numeración de esquinas

Una vez se soluciona este problema, ejecutamos sobre todas las manzanas de la cartografía el algoritmo de numeración de esquinas, el cual, conforme a los requisitos, se encarga de buscar la esquina más al noroeste, y a partir de esta aprovechamos que el paquete *ArcPy*, distribuido en la versión Python integrada en ArcGIS, y en concreto sus objetos iterativos *SearchCursor*, siempre recorren los vértices de las geometrías en orden, para iterar sobre ellos e ir almacenando esa geometrías en una capa a la vez que asignamos una numeración a cada uno de esos elementos generados. El pseudocódigo resultante sería el siguiente:

```
def main():
    # Se itera sobre todas las manzanas de la cartografía
```

```

for manzana in manzanas_cartografia:
    # Se obtiene el envelop o geometría que encierra la manzana.
    envelop = manzana.extent();
    punto_noroeste = (envelop.MaxY, envelop.MaxX)
    for esquina in manzana:
        # Se obtiene la distancia desde la esquina a el punto_noroeste.
        esquinas_distancia.append((esquina.id, distancia_al_punto_noroeste));
        # Se ordena la lista por distancias.
        sorted(esquinas_distancia);
    for esquina_orden in esquinas_distancia:
        # Se introducen en base de datos.
        put_BBDD(esquina_orden, CODESQUINA=num);
        num += 1;

```

El resultado obtenido es una nueva capa de puntos con cada una de las esquinas almacenadas.

4.8 Numeración de lotes

Por último, se deben numerar las parcelas catastrales, que se corresponde con el último punto de los requisitos:

- **[RF1.3]**. Los lotes han tener un código que los identifique de forma única.

La unidad de codificación de los lotes está relacionada con las manzanas que los contienen, siendo lote el término usado en Latinoamérica para referirse a lo que en España se conoce como parcela catastral. Las parcelas catastrales son bienes inmuebles registrados en el Catastro gubernamental, a las que se le asigna una referencia o identificador único. Sus elementos están relacionados con las manzanas que los contienen. Dado que ya se tienen las manzanas numeradas, el código de lote se construye a partir de la numeración secuencial de los lotes en múltiplos de 10 dentro de la manzana. Se adopta el siguiente convenio, de acuerdo con los requisitos del proyecto fijados por el cliente, en concreto el requisito RF1.3, el cual dice: “La codificación de los lotes se realiza de manera similar a la propuesta para las manzanas, arrancando del lote más al norte, en caso de existir varios se debe escoger el que se encuentra en el Noroeste, y enumerarlos en sentido horario empezando por el 0010 en intervalos de 10 hasta el 9990”. Se pueden ver estos requisitos en la Figura 23.



Figura 18. Requisitos de numeración de lotes.

El código de lote tiene el formato SSBMMMLLLL, donde:

- SS: Sector
- B: Subsector
- MMM: Manzana
- LLLL: Lote

De esta manera la primera dificultad que se plantea es la codificación de estos lotes de la cartografía, por lo que se requiere crear un proceso de preprocesamiento de estos elementos.

4.8.1 Codificación de lotes

Al igual que se hizo con la codificación de manzanas, debemos clasificar los lotes o parcelas catastrales por sector, subsector y manzana para un análisis posterior. Para ello esta vez se hace útil la utilización de la función del *SpatialJoin_analysis*, la cual conforme podemos leer en la documentación de *ArcGIS for Desktop* (“Unión espacial—Ayuda | ArcGIS for Desktop,” n.d.) se encarga de “unir los atributos de una entidad con otra basada en la relación espacial. Las entidades de destino y los atributos unidos de las entidades de unión se escriben en la clase de entidad de salida”. En esta ocasión la relación espacial elegida para ejecutar la función de unión es “*HAVE_THEIR_CENTER_IN*”, ya que se prefiere sacrificar la exactitud de las geometrías para su posterior corrección por parte del equipo cartográfico, a cambio de obtener una numeración completa de la capa de lotes.

4.8.2 Algoritmo de numeración de lotes

ArcGIS no permite recorrer los lotes de cada manzana de manera uniforme, es decir, no nos permite recorrer los lotes en sentido horario. Por ello se idea un

algoritmo de numeración eficiente de los lotes. Este, se encarga de recorrer cada uno de los puntos exteriores de los lotes, es decir los vértices de cada uno de los lotes que intersecan con la silueta del conjunto que forma la unión de todos los lotes, como se ve en la Figura 24. Concretamente para cada manzana:

1. Se obtienen todos y cada uno de los vértices de los lotes contenidos en la manzana tratada.
2. Se obtiene la geometría resultante de hacer la unión espacial de todos los lotes de la manzana.
3. De dicha geometría, se calcula la polilínea perteneciente a su perímetro.
4. Se hace la intersección espacial entre el perímetro obtenido y los vértices de los lotes, quedando solo en el conjunto los vértices que intersequen, como se ve en la Figura 25.
5. Se recorren los vértices en sentido horario y empezando por el más al norte, o en caso de existir varios, el más al noroeste, según mandan los requisitos.
6. Se numeran los lotes conformen se vayan encontrando sus vértices correspondientes.

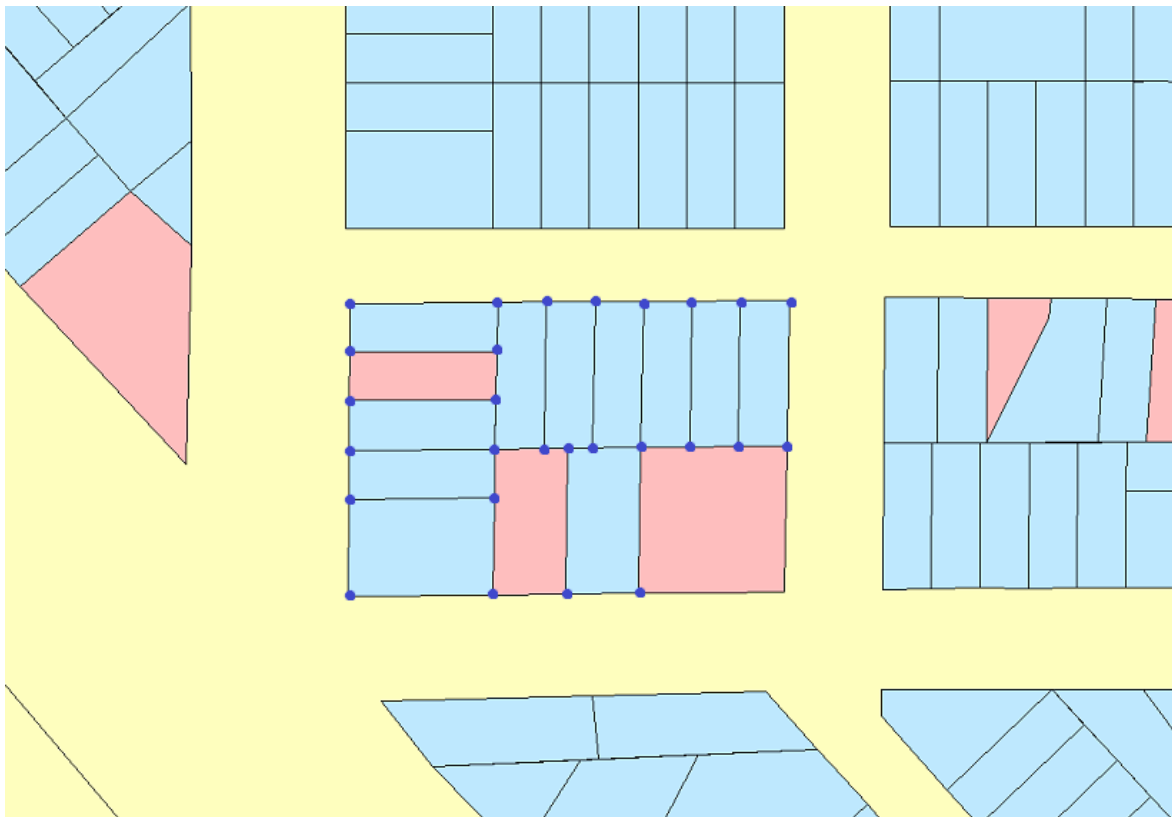


Figura 19. Vértices de lotes que intersecan con la silueta de la manzana.

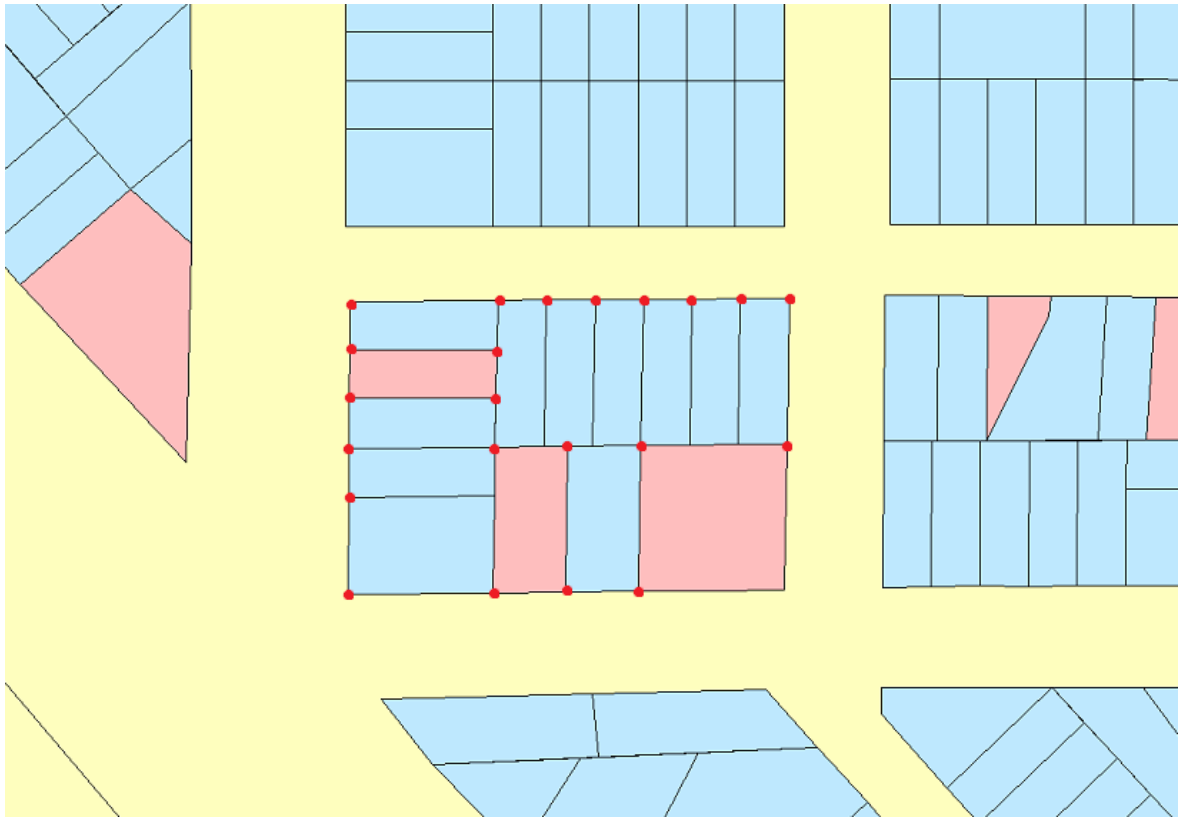


Figura 20. Vértices de lotes que intersecan con el perímetro de la manzana.

El proceso busca recorrer la silueta exterior de la manzana para numerar los lotes imitando una numeración analógica hecha por un humano. El hecho de coger la silueta de la unión de los lotes evita la problemática que se encuentra ante alguna casuística más particular. Este algoritmo se basa en que ArcGIS, y en particular los objetos de iteración de datos del paquete ArcPy (ESRI, n.d.), recorren los vértices de las geometrías en orden. Podemos describir este proceso en pseudocódigo como:

```
def main():
    # Se itera sobre todas las manzanas de la cartografía.
    for manzana in cartografía:
        # Se crea la unión de los lotes.
        for lote in manzana:
            lote_union = lote_union.union(lote);
        # Se obtiene el boundary o perímetro del lote.
        boundary_manzana = lote_union.boundary();
        for lote in manzana:
            for esquina in lote:
                if lote.buffer(2).contains(esquina):
                    # Se ordena la lista de distancias sobre la línea del boundary de la manzana.
                    lista_esquinas.append(boundary_manzana.measureOnLine(esquina));
        # Se ordena la lista de esquinas por distancia sobre la línea.
```

```
# Se ordena la lista de esquinas según la distancia al punto más al norte de la
manzana.
sorted(lista_esquinas);
# Se insertan los lotes codificados en base de datos.
for esquina_orden in lista_esquinas:
    put_BBDD(esquina_orden);
```

A nivel de código se hace uso de algunas funciones nuevas como: *Unión*, método auxiliar de la clase *Polygon* que construye la geometría resultante de la unión espacial de las distintas formas geométricas de entrada; *boundary*, método de auxiliar de la clase *Polygon* que construye el límite de la geometría; *buffer*, un método auxiliar en este caso de la clase *Polyline*, que construye un polígono a una determinada distancia de la geometría, es decir, ensancha la geometría; *contains*, método de análisis de la clase *Polygon* el cual indica si la geometría base contiene a la geometría comparada; *measureOnLine*, método de la clase *Polyline* que devuelve la distancia desde el punto de inicio de la línea hasta el punto pasado por parámetro.

4.9 Utilidades y mejoras

Una vez creados los scripts de numeración se hace necesario, por requisito del proyecto, añadir una sencilla interfaz que muestre los resultados de la ejecución de forma que permita ver si el proceso sigue trabajando o por el contrario se produjo algún error dado que estos procesos son ejecutados por línea de comandos. Para ello se crea la función *print_progress_bar*, contenida en el fichero “utilities.py”, la cual es la encargada de mostrar un *loader* encargado de mostrar al usuario si el proceso aún sigue trabajando a través de la impresión por consola de unos caracteres, como se puede ver en la Figura 26.

```

Run: tecnico_CodificarSubSectores x
(1/8) Progress: |=====| 16.7%
(1/8) Progress: |=====| 20.0%
(1/8) Progress: |=====| 23.3%
(1/8) Progress: |=====| 26.7%
(1/8) Progress: |=====| 30.0%
(1/8) Progress: |=====| 33.3%
(1/8) Progress: |=====| 36.7%
(1/8) Progress: |=====| 40.0%
(1/8) Progress: |=====| 43.3%
(1/8) Progress: |=====| 46.7%
(1/8) Progress: |=====| 50.0%
(1/8) Progress: |=====| 53.3%
(1/8) Progress: |=====| 56.7%
(1/8) Progress: |=====| 60.0%
(1/8) Progress: |=====| 63.3%
(1/8) Progress: |=====| 66.7%
(1/8) Progress: |=====| 70.0%
(1/8) Progress: |=====| 73.3%
(1/8) Progress: |=====| 76.7%
(1/8) Progress: |=====| 80.0%
(1/8) Progress: |=====| 83.3%
(1/8) Progress: |=====| 86.7%
(1/8) Progress: |=====| 90.0%
(1/8) Progress: |=====| 93.3%
(1/8) Progress: |=====| 96.7%
(1/8) Progress: |=====| 100.0% Complete
2019-11-30 19:40:58.735000 execution_time: 0:00:06.257000
2019-11-30 19:40:58.735000 total_subsectors: 30
2019-11-30 19:40:58.735000 worked_subsectors: 30

```

Figura 21. Salida por consola de la ejecución del script de numeración de subsectores.

Otra de las mejoras que se hacen por necesidades del proyecto, es la inclusión de parámetros en todos los scripts para permitir el filtrado de los sectores, subsectores, manzanas y lotes a numerar. Esto es debido a la necesidad de mantener la numeración de los elementos ya entregados al cliente, ya que la alteración de esta provocaría la repetición de las tareas en campo. Para ello se añaden los siguientes parámetros:

- (-s): Sector o lista de sectores sobre el que se ejecuta el script. p. ej. -s '10', '2'
- (-sb): Subsector o lista de subsectores sobre el que se ejecuta el script. p. ej. -sb '1', '2'
- (-b): Manzana o lista de manzanas sobre la que se ejecuta el script. p. ej. -b '101', '102', '103'
- (-o): Orden de numeración de las esquinas. (N | NE | NO). p. ej. -o N

Por otro lado, se dota a los scripts de un análisis temporal de sus ejecuciones monitorizando el tiempo que tardan en ejecutarse. Esto permite hacer un examen

exhaustivo de la ejecución permitiendo valorar los cambios que se van introduciendo. Esto se hace a través de las librerías estándar de *Python*, en concreto se usa el módulo *time*, el cual ofrece métodos avanzados que permiten trabajar estas unidades de tiempo (Python, n.d.).

4.10 Resultados

El producto que obtenemos tras la ejecución de los procesos automáticos de numeración de los diferentes elementos cartográficos es una cartografía referenciada y preparada para ser usada en las diferentes tareas a llevar a cabo en campo por los operarios técnicos del cliente. Podemos ver el resultado final en las figuras que aparecen a continuación. Estas han sido generadas gracias al visor de ArcMap en ellas se muestran de las etiquetas de referencia encima de cada uno de los elementos.

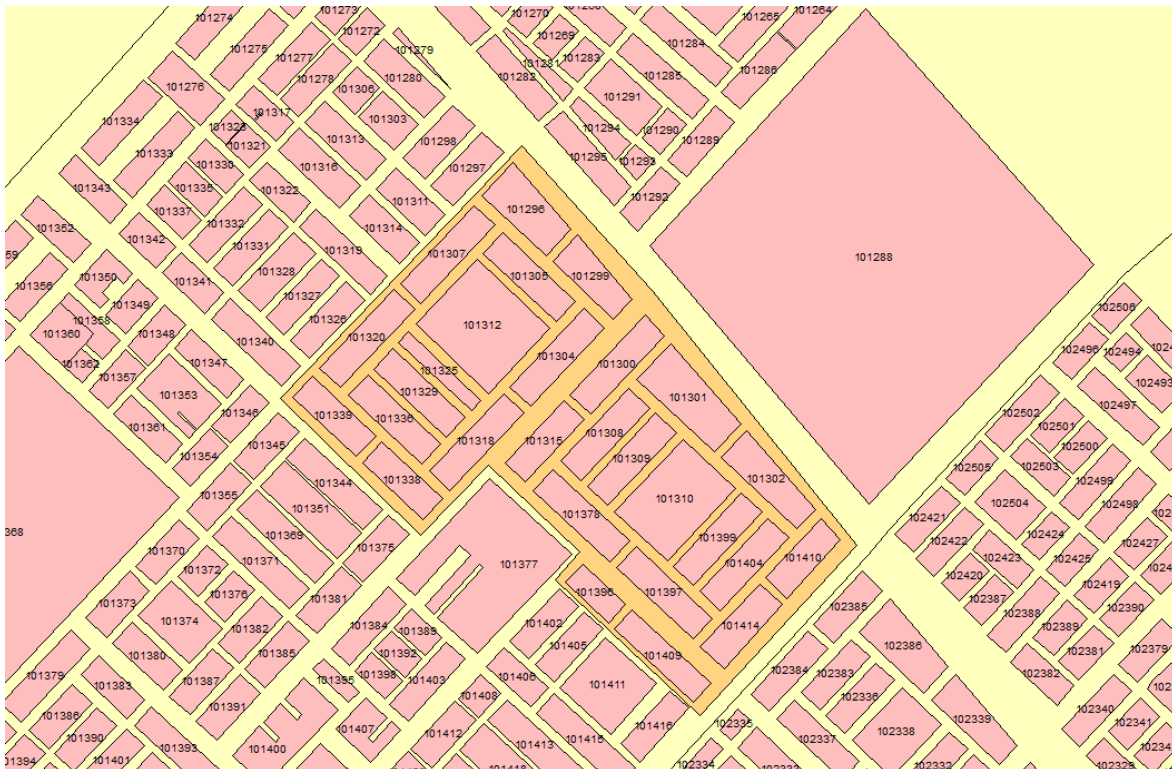


Figura 22. Manzanas numeradas tras el proceso de codificación de la cartografía base.

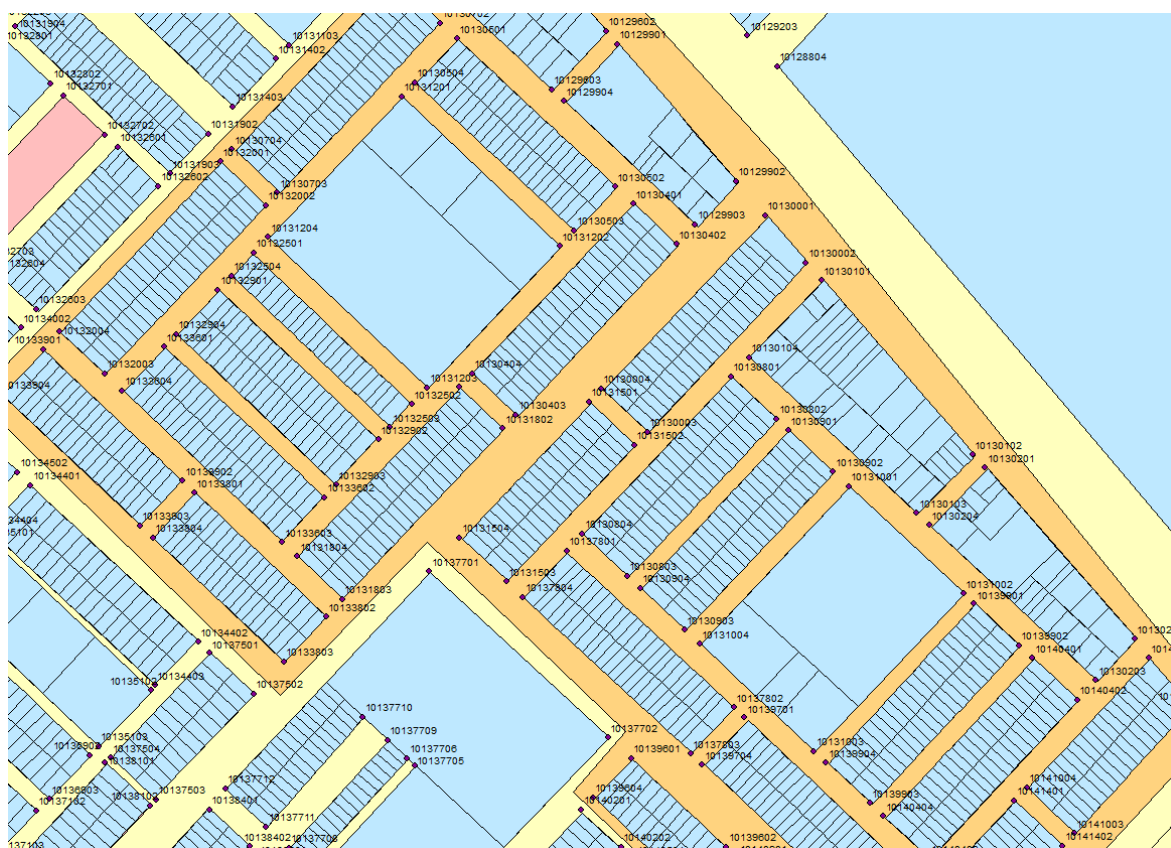


Figura 23. Esquinas numeradas tras el proceso de codificación de la cartografía base.

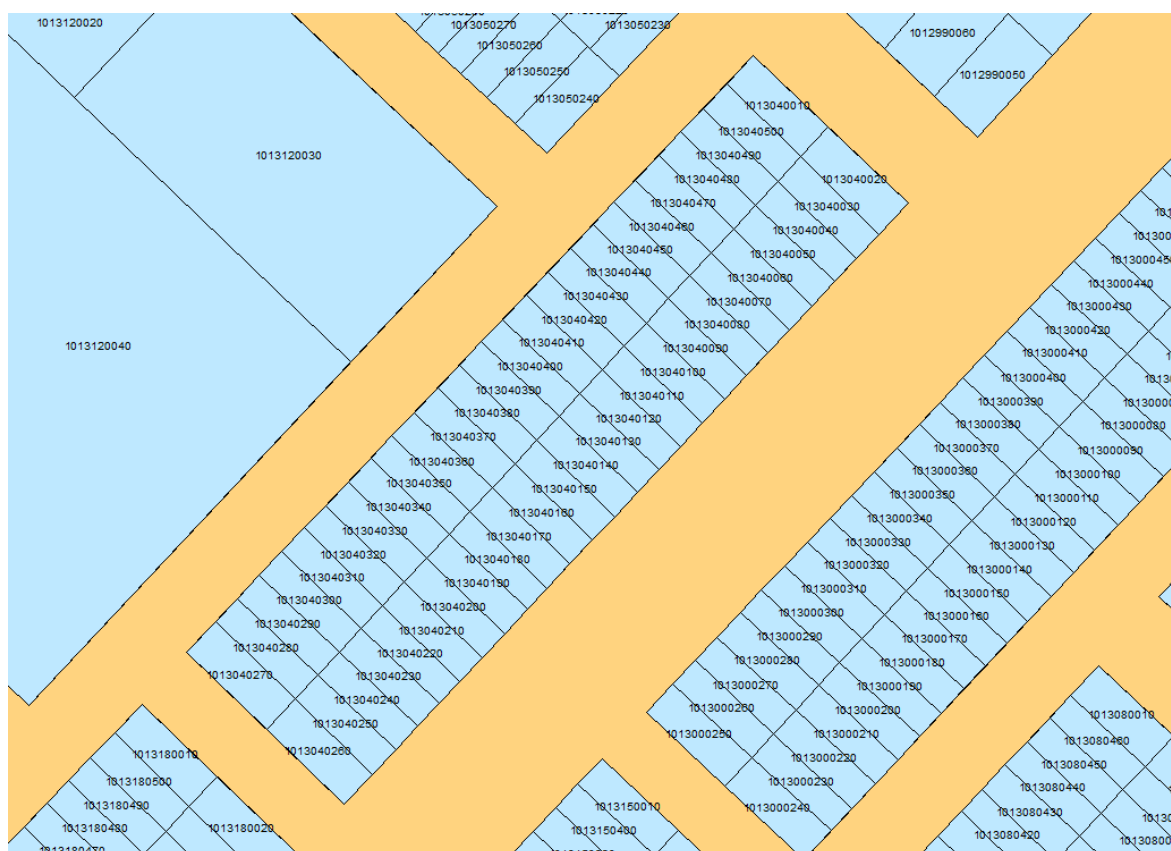


Figura 24. Lotes numerados tras el proceso de codificación de la cartografía base.

4.11 Conclusiones

En este capítulo se ha completado la codificación y numeración de todos los elementos de la cartografía base, entendiendo la codificación como el proceso con el cual ubicamos un elemento asignándole como atributos los códigos de todos los elementos que lo contienen. Por otro lado, entendemos la numeración como el proceso de asignación de un número que identifique el elemento de manera única e inequívoca. Para ello previamente se ha presentado la cartografía base a numerar, y se ha hecho un estudio de necesidades valorando las distintas tecnologías candidatas de ser implementadas. Por último, se ha hecho un análisis de los algoritmos de numeración implementados y se han valorado los resultados obtenidos.

Capítulo 5

Sistema de clasificación de elementos cartográficos

5.1 Introducción

En este capítulo se documentan los trabajos del último de los subobjetivos para la construcción de un sistema de clasificación de elementos de la cartografía base. Para ello se realiza un estudio de necesidades inicial en el que se analizan y comparan distintas tecnologías candidatas de ser usadas para la construcción del sistema. Además, se define y crea la base de conocimiento que se usará para el experimento. A continuación, se documentan los trabajos para la implementación del sistema y, por último, se analizan los resultados obtenidos, apoyando las conclusiones en distintas representaciones gráficas de los datos obtenidos.

5.2 Estudio de necesidades


La digitalización que se obtuvo por parte del cliente no es del todo correcta, y mucho menos precisa, por lo que el trabajo de comprobación de los elementos y corrección de estos a través de una nueva digitalización de los errores encontrados se hizo arduo e interminable de acuerdo con los plazos del proyecto. Uno de los principales problemas con el que se encuentra el proyecto es la corrección y remodelación de los lotes contenidos en las manzanas de la cartografía base. Estos se encuentran en muchas ocasiones mal ubicados, con




su posición original movida o no corresponden con las figuras de las parcelas catastrales representadas. Este problema se resolvió por el equipo de cartógrafos especializados en informática que componen el equipo de trabajo de la empresa. El equipo emplea fotografías satelitales u ortofotos para corregir dichos errores apoyándose en la utilización tanto del SIG de ArcGIS como otros programas de modelado más específicos. Sin embargo, dado el coste temporal de esta tarea, se intenta automatizar dicho proceso o, al menos, acelerar el proceso y con ello los tiempos del proyecto.

Los trabajos necesarios para completar la codificación de la cartografía se estiman en tiempo muy por encima de lo imaginado al inicio del proyecto. Esto hace que se busquen soluciones que, permitan reducir tiempo y complejidad de las tareas al equipo cartográfico del proyecto. Por ello, y con el fin de reducir estos tiempos del proyecto, se inició la construcción de un sistema que, apoyado en la inteligencia artificial, en concreto en el aprendizaje automático (ver Sección 2.4), sea capaz de clasificar algunos tipos de manzanas para posteriormente automatizar el proceso de digitalización de sus lotes.

Lo primero que se necesita es la creación de una base de datos en la que almacenar las características de cada manzana. Para ello barajamos varias opciones o sistemas con los que trabajar, y realizamos un breve estudio o análisis previo con el que decidir cuál de los motores se ajusta mejor a nuestras necesidades. Por ello, ponemos a estudio los siguientes motores de base de datos:

Tabla 15. Estudio de necesidades de la base de datos.

	Características				
	Compañía	Licencia	Lenguaje de consulta	Documentación	Integración
MySQL 	Oracle	Trial	SQL	Aceptable	Fácil
Postgre SQL	PostgreSQL	Gratis	PL PgSQL	Excelente	Fácil

					
Oracle 	Oracle	Trial	PL SQL	Aceptable	Fácil
Access 	Microsoft	Gratis (UCLM)	SQL	Excelente	Moderada

En este caso se elige crear la base de datos utilizando *PostgreSQL* (PostgreSQL, 2019), en concreto la versión 11 debido a que incorpora las últimas novedades hasta la fecha. Se toma esta decisión debido a que *PostgreSQL* es un software open source, gratuito y con una gran comunidad activa donde resolver las dudas que podamos ir encontrando. Además, la curva de aprendizaje con respecto a otros motores de base de datos como *SQL Server* u *Oracle* va a ser mucho menor debido al contacto previo que se tuvo en asignaturas del grado relativas a la rama del conocimiento de tratamiento y almacenamiento de datos. También debido a las bajas necesidades respecto a los datos a almacenar, contando solo con datos binarios, no se hace necesaria la utilización de ningún sistema de base de datos con capacidad de análisis espacial. Sin embargo, de haber sido necesario, existen alternativas basadas en el motor elegido *PostgreSQL*, como son *PostGIS* (PostGIS, 2019), extensión de la base de datos para el manejo de datos espaciales.

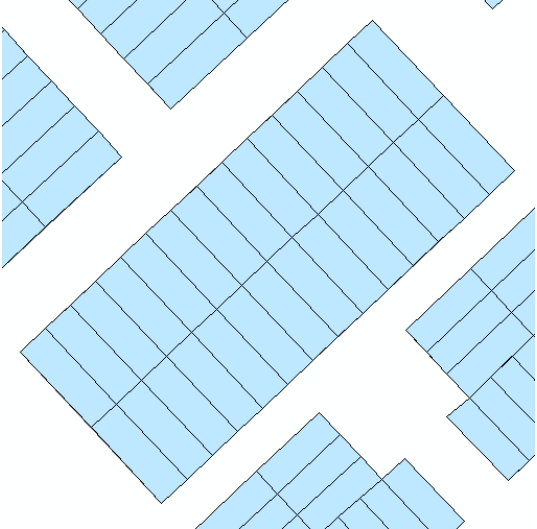
Una vez realizada la base de datos necesitaremos apoyarnos en algún software o librería con la que hacer ese estudio automático de los datos que nos permita clasificarlos. Esta vez se apuesta por una solución en la nube, Microsoft Azure Machine Learning Studio (Microsoft, 2019b). Ésta es una herramienta de Microsoft que nos permite reutilizar algoritmos inteligentes de manera sencilla, con un proceso muy simple de drag and drop. Lo que nos hace decantarnos por esta opción antes que usar otras librerías más extendidas como Scikit-learn para Python (Scikit learn, 2019b) o Dlib en C++ (Dlib, 2019a), es su facilidad de uso gracias a su interfaz usable, y un grid en el que posicionar diferentes elementos o procesos representados por cajas. Otra ventaja es su facilidad para compartir

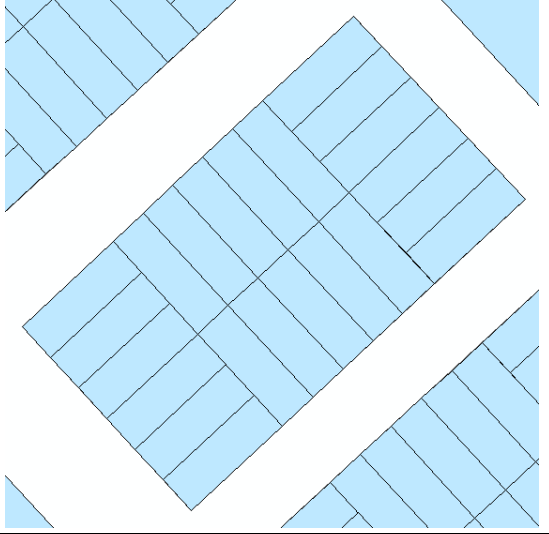
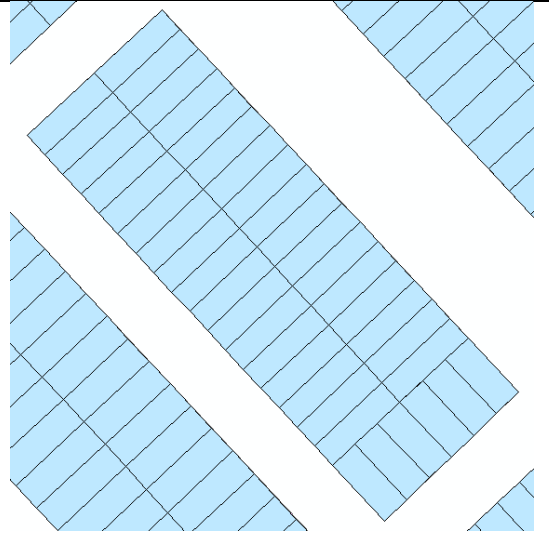
resultados lo que facilita la muestra de resultados una vez ejecutado el experimento. También valoramos la ventaja de usar software comercial, desarrollado por Microsoft, debido a que su uso se encuentra ampliamente probado y estandarizado.

5.3 Tipos de Manzanas

Los trabajos de construcción del sistema empiezan por identificar algunos tipos de manzanas que se repitan con frecuencia en la cartografía y cuyo proceso de corrección vaya a ser llevado a cabo varias veces. Para ello, el equipo de cartografía identifica los tipos de manzana que más dificultades están causando en el proceso. Estos tipos, aunque no solucionan todos los problemas ofrecidas por la cartografía, sí cubren gran parte de los problemas existentes. Dichos tipos son los siguientes:

Tabla 16. Tipos de manzanas predefinidas por observación directa.

Nombre	Forma	Previsualización
TIPO A	$2 \times N$	

TIPO B	Parte oeste: $1 \times N_1$ Parte central: $2 \times N_2$ Parte este: $1 \times N_1$	
TIPO C	Parte lateral: $1 \times N_1$ Parte central: $2 \times N_2$	

5.4 Construcción de la base de conocimiento

Antes de la construcción de nuestro sistema, es necesario procesar los datos espaciales con los que contamos a fin de extraer nuevas características y variables útiles para el proceso de aprendizaje automático que se llevará a cabo. Para ello se construye una base de datos sobre *PostgreSQL*, tal y como se justificó en la Sección anterior, con datos binarios obtenidos de los datos espaciales de la cartografía base. Esta base de datos, únicamente formada por una tabla, constituirá la base de conocimiento o *dataset* con el que nuestro algoritmo trabajará.

Debido a los problemas con los que cuenta la cartografía inicial, se decide trabajar con un subconjunto menor de muestras de elementos de la cartografía ya corregidos y ajustada su posición por el equipo de cartografía. Gracias a esto,

aunque perdemos muestras con las que trabajar, conseguimos una mayor fiabilidad de los datos.

En el CD que acompaña a este TFG, se encuentran todos y cada uno de los scripts SQL necesarios para la reconstrucción y reproducción de la base de datos utilizada, así como la misma base de datos finalmente construida. Un resumen de los atributos que describen cada elemento cartográfico de las manzanas se presenta en la tabla siguiente:

Tabla 17. Definición de variables de la base de conocimiento.

	Variable	Descripción	Tipo
1	c1_plots_number	Número de lotes con salida en C1.	<i>Integer</i>
2	c2_plots_number	Número de lotes con salida en C2.	<i>Integer</i>
3	l1_plots_number	Número de lotes con salida en L1.	<i>Integer</i>
4	l2_plots_number	Número de lotes con salida en L2.	<i>Integer</i>
5	x_block_coordinate	Coordenada x de la manzana.	<i>Real</i>
6	y_block_coordinate	Coordenada y de la manzana.	<i>Real</i>
7	objectid	Identificador único de la manzana en el sistema <i>ArcGIS</i> .	<i>Integer</i>
8	codmanzana	Código de identificación de la manzana dentro del subsector.	<i>Integer</i>
9	codsector	Código de identificación del sector.	<i>Integer</i>
10	codsubsector	Código de identificación del subsector dentro del sector.	<i>Integer</i>
11	x_centroid_coordinate	Coordenada x del centroide de la manzana.	<i>Real</i>
12	y_centroid_coordinate	Coordenada y del centroide de la manzana.	<i>Real</i>

13	x_true_centroid_coordinate	Coordenada x del centroide verdadero de la manzana.	<i>Real</i>
14	y_true_centroid_coordinate	Coordenada y del centroide verdadero de la manzana.	<i>Real</i>
15	block_area	Área de la manzana.	<i>Real</i>
16	block_length	Longitud de la manzana.	<i>Real</i>
17	c1_length	Longitud del lado C1.	<i>Real</i>
18	c2_length	Longitud del lado C2.	<i>Real</i>
19	l1_length	Longitud del lado L1.	<i>Real</i>
20	l2_length	Longitud del lado L2.	<i>Real</i>
21	block_apexes_number	Número de vértices de la manzana.	<i>Integer</i>
22	rectangular_form	¿Manzana con forma rectangular?	<i>Boolean</i>
23	square_form	¿Manzana con forma cuadrada?	<i>Boolean</i>
24	is_multipart	¿La figura posee varias partes?	<i>Boolean</i>

Para la obtención de datos se construye un script en *Python*, también adjunto a este trabajo, por nombre *spider_database_data.py*. El script obtiene, apoyándose en los métodos y funciones del paquete *ArcPy*, todos los valores relativos a las características de la base de datos. Una vez creada la base de datos y ejecutados los scripts que inyectan los datos en ella, se decide filtrar las muestras disponibles, para dejar solo los elementos cartográficos de manzanas que contengan únicamente 4 vértices. Por ello se eligió previamente trabajar con una cartografía ya corregida que evitase que las manzanas tuvieran lados extra mal codificados que evitarían elegir manzanas que, aunque tengan forma rectangular, y por lo tanto cuatro vértices reales, por una mala codificación posean más. Para ello, se crea una vista con este subconjunto de muestras para más tarde exportar dicho conjunto de datos como fichero CSV, el cual es el formato aceptado como una fuente de datos en *MAMLS*.

5.5 Algoritmo

Para la elección del algoritmo adecuado a utilizar se ha empleado la *Cheat Sheet* o chuleta de la biblioteca *Scikit-learn* (Scikit learn, 2019a). Esta hoja permite

decidir el algoritmo adecuado para resolver el problema a tratar respondiendo unas sencillas preguntas que te guían en un camino de decisión tal y como se puede apreciar en la figura siguiente.

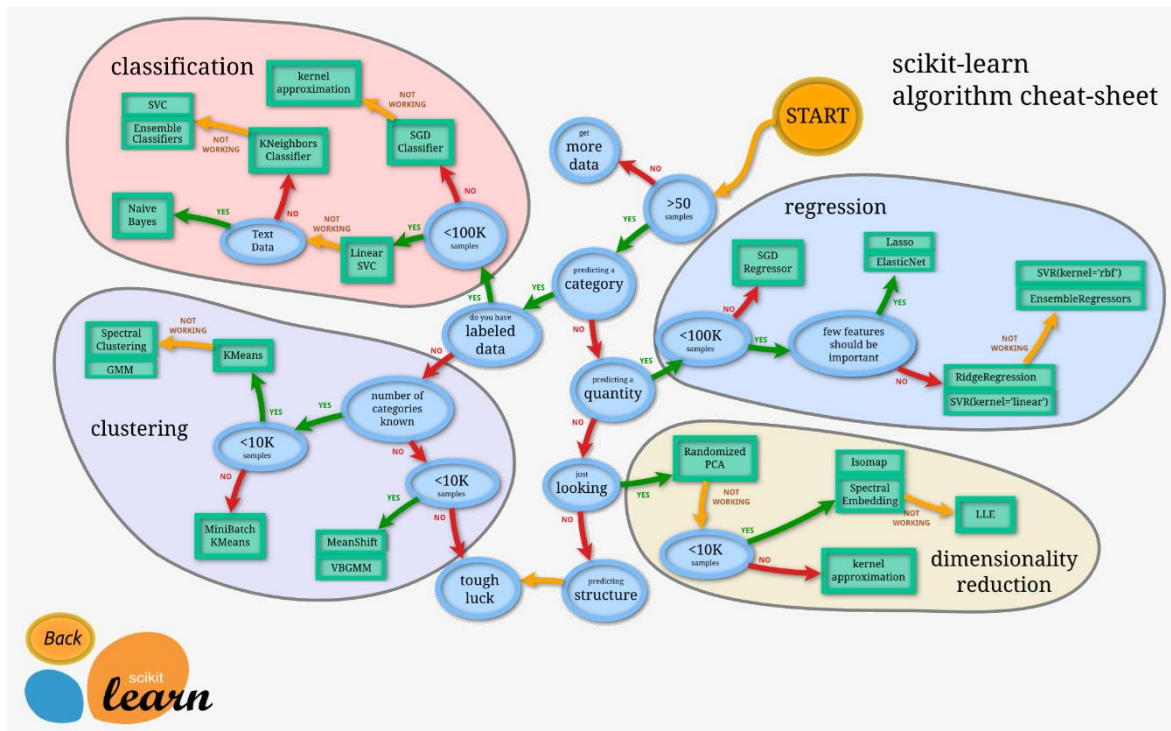


Figura 25. Cheat Sheet creado por Scikit Learn.

Existen otros ejemplos de hojas de ayuda de este tipo, como la creada por los autores de la librería *Dlib* (Dlib, 2019b) (ver Figura 31) de aprendizaje automático para C++, o la creada por Microsoft Azure Machine Learning (Microsoft, 2019a). Hemos decidido utilizar Scikit Learn porque es la que más claramente establece el algoritmo a implementar una vez recorrido el camino de decisión de la chuleta.

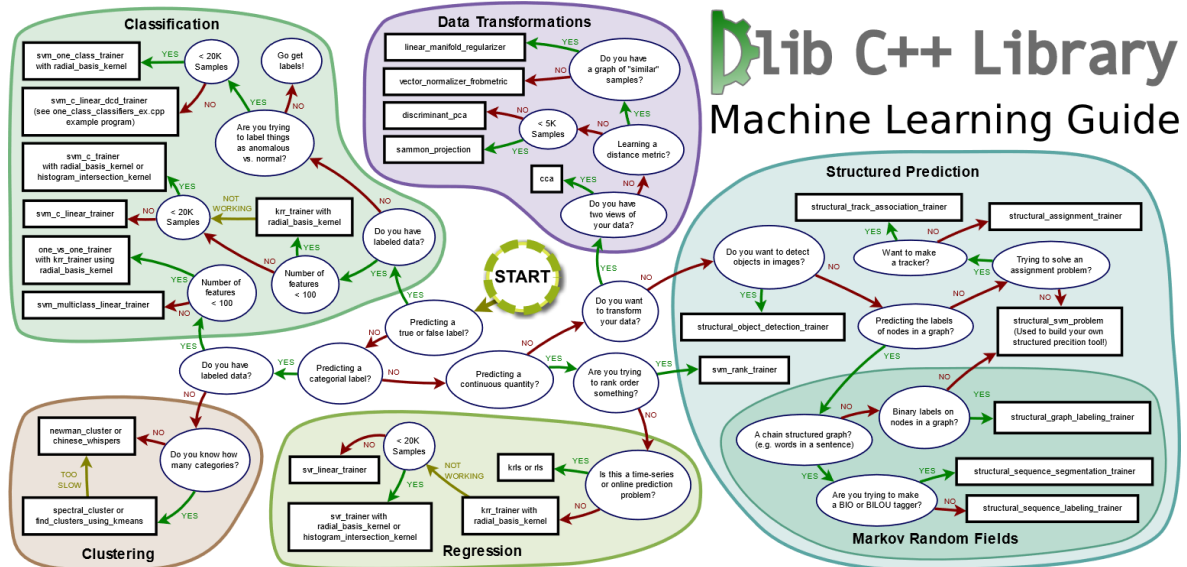


Figura 26. Cheat Sheet creado por Dlib.

Aplicando la guía descrita por la hoja de ayuda de *Scikit Learn*:

- Más de 50 muestras (> 50 samples): Sí
- Predice una categoría (Predicting a category): Sí
- ¿Posees datos etiquetados? (Do you have labeled data): No
- Número de categorías conocido (Number of categories known): Sí
- Menos de 10.000 muestras (< 10K samples): Sí

El camino arroja como resultado la recomendación de uso del algoritmo *KMeans* (Tiwari, 2017), y en caso de no funcionar adecuadamente recomienda usar el algoritmo *Spectral Clustering GMM* (Tiwari, 2017). *KMeans* es uno de los algoritmos de aprendizaje automático más sencillos y populares. Se conoce como un algoritmo de clasificación no supervisada. Su objetivo es simple: Agrupar puntos similares juntos descubriendo relaciones entre ellos. Para ello busca en un grupo fijo (k) de *clústeres* en un conjunto de datos.

El algoritmo realiza dos tareas:

- **Inicialización:** El algoritmo escoge k puntos según la estrategia elegida, los cuales son usados como centroides de cada *clúster*. Normalmente la estrategia inicial suele ser elegir los puntos aleatoriamente.
- **Actualización de los centroides:** El algoritmo itera en este paso para actualizar la posición de los centroides con el promedio de los objetos que pertenecen a dicho grupo. El algoritmo se detiene una vez que la nueva posición de los centroides no se ha visto alterada por la última iteración o que se mantiene por debajo de un umbral dado.

KMeans resuelve por medio de la optimización el problema, mejorando el valor de la suma de las distancias cuadráticas al centroide al que pertenece cada punto. Los puntos se representan como vectores reales (x_1, x_2, \dots, x_n) , agrupándolos en k *clústeres* o grupos $S = \{S_1, S_2, \dots, S_k\}$. Por lo que formulamos el problema como:

$$\min_S E(\mu_i) = \min_S \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|^2$$

Donde S representa el conjunto de datos o *dataset*, con elementos vectores x_j ., existiendo k *clústeres* con sus correspondientes centroides μ_i .

Las grandes ventajas del algoritmo de clasificación aquí explicado *KMeans* son su sencillez de uso y su rapidez a la hora de ofrecer una solución. Sin embargo, la parametrización de k y sus diferentes hiperparámetros, hace difícil su rendimiento hacia la búsqueda de un óptimo, haciendo que el algoritmo llegue a converger a mínimos locales.

5.6 Construcción del sistema

Hoy en día existen diversos servicios, tanto en la nube como en local, que nos permiten construir un proceso de aprendizaje automático de manera sencilla. Todos ellos cuentan con diferentes ventajas que los caracterizan y poseen algunas similitudes comunes como son la escalabilidad del sistema, el gran abaratamiento de los costes que han sufrido la mayoría de estos servicios en los últimos años y la sencillez de uso que nos ofrecen. Al tratarse de un mercado demasiado amplio, en la siguiente Sección se estudian los más conocidos y que mejor se adaptan al diseño de la solución, permitiendo el manejo de grandes volúmenes de datos y una fácil reproducción de los experimentos para poder depurar los resultados. Con el resultado de dicho estudio, en la Sección.5.2 se describe la implementación realizada.






5.6.1 Análisis de alternativas

Para elegir la mejor alternativa para la construcción del sistema elegimos algunas de las plataformas y programas disponibles actualmente para el cómputo de soluciones de Machine Learning. Entre ellas encontramos la plataforma Microsoft Machine Learning Studio de la compañía Microsoft, la plataforma

donde el usuario es capaz de trabajar con la ciencia de los datos a través de análisis predictivos en la nube de los datos a través de la construcción de experimentos configurando diferentes bloques en un grid (Barga, Fontama, & Tok, 2015). Otra plataforma muy parecida, pero de la compañía Google, es Cloud Machine Learning Engine, la cual ha tenido un gran crecimiento entre la comunidad de desarrolladores en los últimos años. También analizamos TensorFlow, la solución que permite, a través de diferentes lenguajes de programación, desarrollar y entrenar diferentes modelos y desplegar las soluciones creadas en diferentes plataformas. Por último, se comparan también las plataformas de Deep Learning Studio y Amazon SageMaker, la solución en la nube de Amazon para la creación de soluciones de Machine Learning.

Para una mejor y más visual comparación se propone una tabla comparativa entre algunos de los servicios elegidos:

Tabla 18. Comparativa de servicios Machine Learning en la nube.

	Machine Learning Studio	Cloud Machine Learning Engine	TensorFlow	Deep Learning Studio	Amazon SageMaker
Compañía					
Coste	Gratis	Licencia 12 meses	Gratis	Gratis	Licencia gratuita con la capa de AWS
Documentación	Excelente	Excelente	Excelente	Correcta	Correcta
Soporte del algoritmo utilizado	Si	Si	Si	Si	Si
Metodología de uso	Canvas, Módulos	Canvas, Integración de servicios	Código API	Canvas	Integración de servicios

Tras este análisis se decide empezar los trabajos utilizando la plataforma de Microsoft Azure Learning Studio debido a la sencillez de construcción de nuevas soluciones que tiene gracias a su sistema de drag and drop en el canvas

interactivo y la amplia documentación disponible sobre la plataforma. Además, se valora que la universidad cuenta con licencia para su uso, permitiendo a los estudiantes investigar de manera gratuita.

5.6.2 Implementación de la solución

Para comenzar el desarrollo se accede a la página principal de Microsoft Azure Machine Learning Studio. Una vez dentro se procede a crear un nuevo proyecto desde la pantalla principal del portal en la sección *Projects*, como se puede ver en Figura 27.

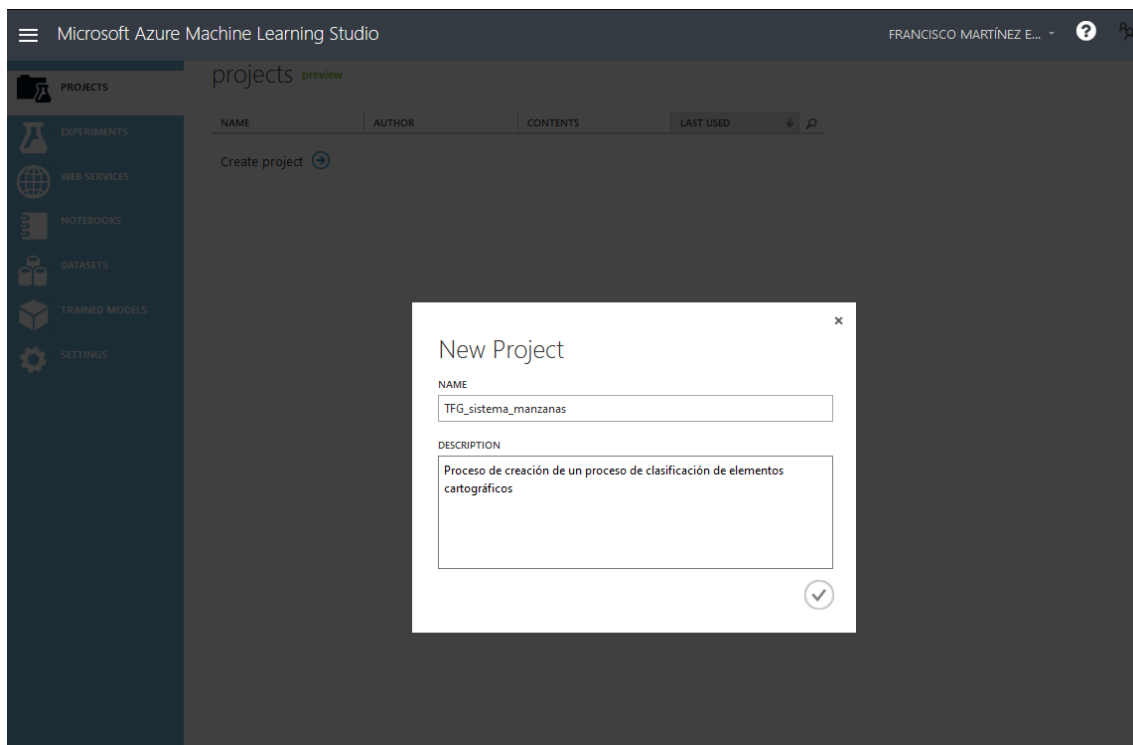


Figura 27. Creación de un nuevo proyecto en MAMLS.

En el siguiente paso, se carga el fichero CSV, exportado de la base de datos *PostgreSQL*, donde se ha almacenado toda la base de conocimiento para analizar (ver 5.4). Este fichero se carga en la sección *DATASETS* de *MAMLS*, introduciendo la ruta o *path* del fichero CSV, el nombre identificativo del dataset y seleccionando la opción *Generic CSV File with a header*, con el que indicamos que nuestro *dataset* contiene el nombre de las variables en la primera fila.

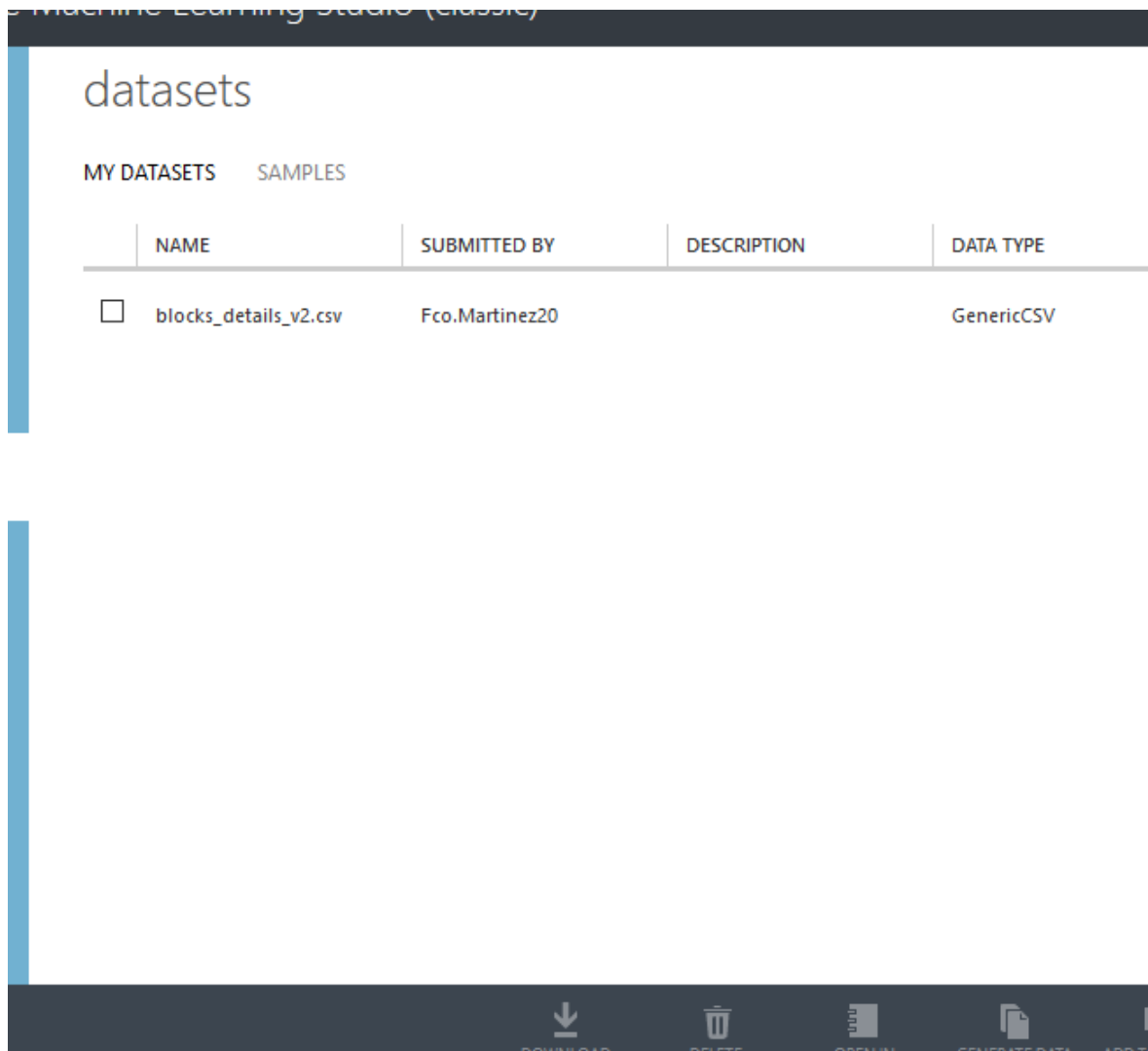


Figura 28. Base de conocimiento creada en MAMLS.

A continuación, se crea un nuevo experimento desde el apartado *Experiments* de *MAMLS*. Para ello se utiliza la opción *New* y en este caso se decide usar la plantilla *Blank Experiment* o experimento en blanco. Tras ello, se procede a diseñar el sistema de agrupamiento *K-Means Clustering* arrastrando los bloques necesarios al canvas de trabajo, que son (ver Figura 29):

- *blocks_details_v2.csv*: Fuente de datos asociada al dataset subido previamente.
- *Select Columns in Dataset*: Módulo previo a la computación del modelo que permite hacer un preprocesado del dataset eligiendo solo las columnas que desee el usuario. Módulo que devuelve un dataset con las columnas del dataset seleccionadas.
- *K-Means Clustering*: Módulo del algoritmo inicial de *clustering* usado en el entrenamiento.

- *Train Clustering Model*: Módulo encargado del entrenamiento del modelo a partir de los datos aportados en el dataset de entrada.
- *Select Columns in Dataset*
- *Convert to CSV*: Módulo encargado de convertir el dataset resultado en un fichero CSV para su exportación y posterior tratamiento en *ArcGIS*.

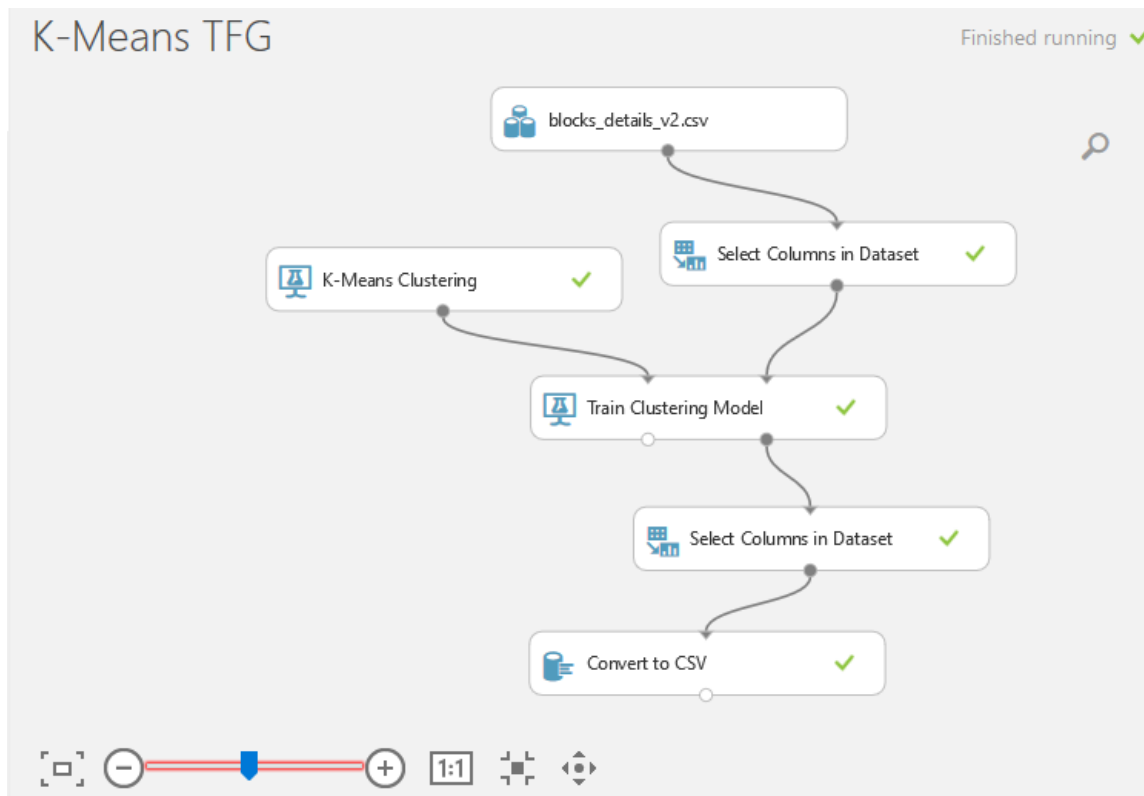


Figura 29. Esquema del experimento de clustering en MAMLS

Una de las características de *MAMLS* es la posibilidad de parametrización que brinda, de muchos de sus módulos, entre ellos el módulo *K-Means Clustering* usado en este experimento. Entre sus posibles opciones encontramos las siguientes:

Create trainer mode: Especifica cómo será entrenado el modelo.

- *Single Parameter*: Se utiliza en caso de conocer el número exacto de parámetros para usar en el modelo de *clustering*.
- *Parameter Range*: Se utiliza si no se conocen los parámetros exactos por lo que se usa el módulo *Sweep Clustering* para obtener la configuración óptima.

Number of centroids: Número de clústeres con los que empezará el algoritmo. Esto no asegura que la configuración final vaya a tener este número de clústeres.

Initialization: Algoritmo utilizado para definir la configuración inicial de los clústeres.

- *First N*: Son elegidos algunos puntos y se utilizan como las medias iniciales.
- *Random*: Los centroides son elegidos al azar.
- *K-Means++*: Es una mejora del algoritmo *K-means* usando un método diferente para elegir los centros iniciales de los centroides (Arthur & Vassilvitskii, 2007).
- *K-Means++Fast*: Una variedad del anterior optimizada en rendimiento y tiempo.
- *Evenly*: Los centroides elegidos son equidistantes entre sí en el espacio de datos.
- *Use label column*: Se elige los centroides iniciales inducidos por la variable clase o etiqueta.

Random number seed (Opcional): Valor que inicializa la semilla usada en la inicialización de los clústeres.

Metric: Distancia usada para medir distancias entre clústeres.

- Euclidean
- Cosine

Iterations: Número de veces que iterará el algoritmo sobre el conjunto de entrenamiento.

Assign label mode: Forma de inserción de la variable clase en el modelo.

- Ignore label column
- Fill missing values
- Overwrite from closest to center

A la hora de entrenar nuestro modelo, elegimos la configuración descrita en la siguiente Tabla:

Tabla 19. Configuración para la creación del modelo MAMLS.

Variable	Configuración elegida
Create trainer mode	<i>Single Parameter</i>
Number of centroids	4
Initialization	<i>K-Means++</i>
Random number seed (Opcional)	
Metric	<i>Euclidean</i>

<i>Iterations</i>	100
<i>Assign Label Mode</i>	<i>Ignore label column</i>

Esto se debe a que, en este caso, se desea utilizar todas las variables posibles que brinden información diferencial de cada una de las manzanas, debido a la poca información que se tiene de ellas. Además, se desea evitar inicializar el modelo únicamente con el valor de los lotes que posee la manzana en cada lado. El número de centroides inicial es cuatro: tres para los centroides de los casos diferenciados en la Sección 5.3, y uno más para casos no concluyentes del modelo. Se decide también ignorar la variable clase, ya que previo a la ejecución del algoritmo no se conoce y se itera un total de cien veces en la inicialización. Tras la creación del experimento en *Microsoft Azure Machine Learning Studio* se obtiene un dataset formado por las siguientes variables:

Tabla 20. Variables del dataset de salida tras ejecutar el algoritmo de clustering.

	Variable	Descripción	Tipo
1	<i>c1_plots_number</i>	Número de lotes con salida en C1.	<i>Integer</i>
2	<i>c2_plots_number</i>	Número de lotes con salida en C2.	<i>Integer</i>
3	<i>l1_plots_number</i>	Número de lotes con salida en L1.	<i>Integer</i>
4	<i>l2_plots_number</i>	Número de lotes con salida en L2.	<i>Integer</i>
5	<i>DistanceToClusterCenter no.0</i>	Variación respecto al <i>clúster</i> n.0	<i>Real</i>
6	<i>DistanceToClusterCenter no.1</i>	Variación respecto al <i>clúster</i> n.1	<i>Real</i>
7	<i>DistanceToClusterCenter no.2</i>	Variación respecto al <i>clúster</i> n.2	<i>Real</i>
8	<i>DistanceToClusterCenter no.3</i>	Variación respecto al <i>clúster</i> n.3	<i>Real</i>
9	<i>objectid</i>	Identificador único de la manzana en el sistema <i>ArcGIS</i> .	<i>Integer</i>
10	<i>assignments</i>	Variable clase que identifica el <i>clúster</i> al que corresponde dicha manzana.	<i>Integer</i>





Gracias al módulo de exportación de los datos a formato CSV, se puede cómodamente trabajar con los resultados fuera de la plataforma. Se crea

entonces un *script* básico escrito en *Python* que nos permite la unión de la información de etiquetado de cada una de las manzanas, con su información geográfica almacenada en la *geodatabase*, tratada anteriormente en la Sección 5.2, propia de *ArcGIS*.

5.7 Resultados

Una vez se ejecuta el experimento creado en la plataforma de *Microsoft Azure Machine Learning Studio*, el resultado que se obtiene es un modelo entrenado preparado para ser utilizado en futuras ejecuciones, así como la base de datos clasificada de elementos cartográficos de manzanas clasificadas por los tipos buscados. Gracias al visor de *ArcMap*, se realiza un mapa temático de esta clasificación por tipos de manzanas creados. El resultado, aunque no es totalmente preciso, sí que ayuda enormemente de cara a poder agilizar los trabajos de remodelación de la cartografía base. Para una mejor comprensión del mapa, y a modo de leyenda, se utiliza la leyenda de colores descrita en la siguiente Tabla:

Tabla 21. Leyenda del mapa temática que muestra los resultados de la clasificación.

Tipo de manzana	Color de representación	Código <i>rgb</i>
A		<i>rgb(252, 179, 250)</i>
B		<i>rgb(179, 252, 182)</i>
C		<i>rgb(182, 207, 252)</i>
Otras		<i>rgb(225, 225, 225)</i>

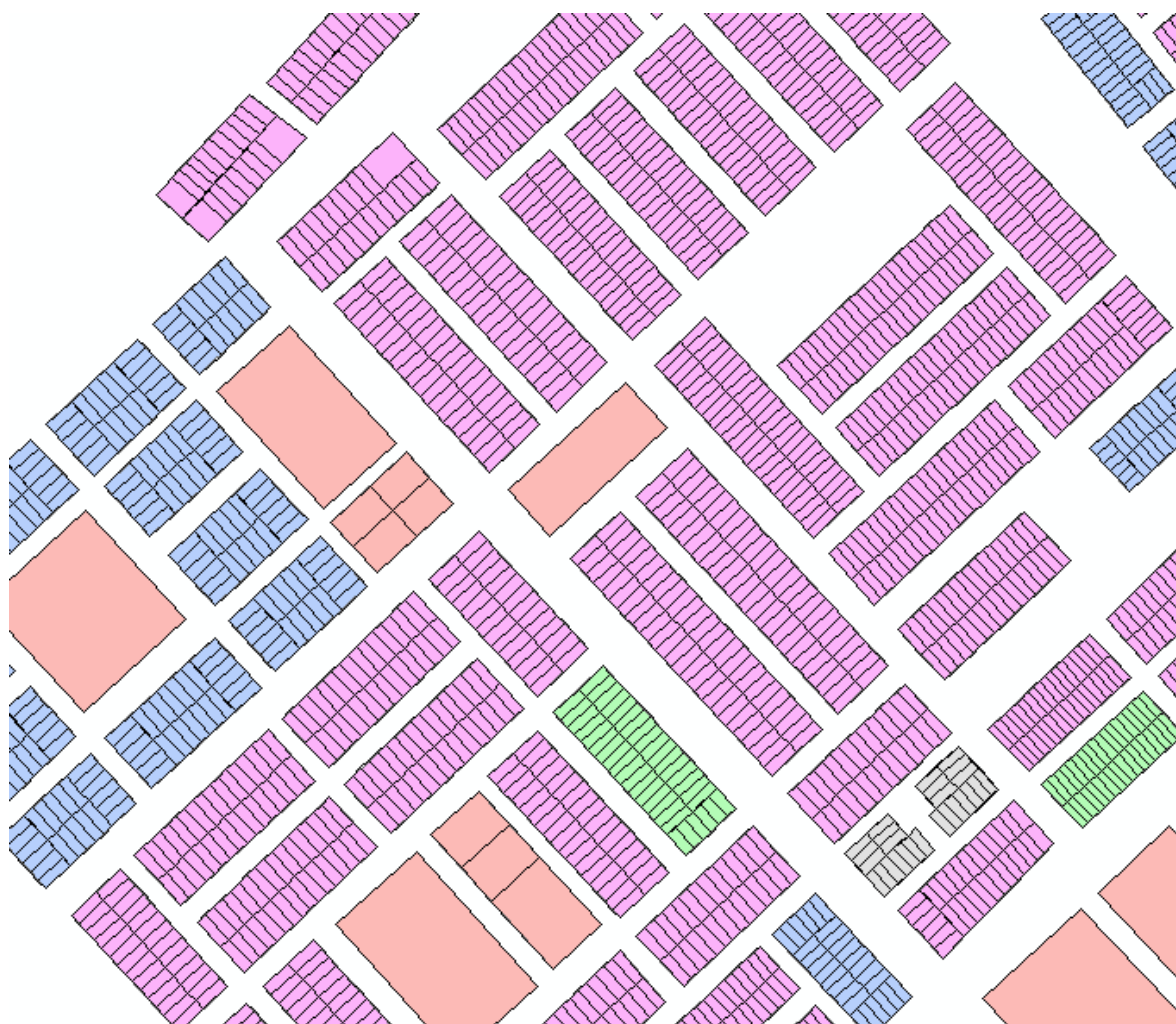


Figura 30. Mapa temático del resultado de la clasificación de la cartografía.

Para valorar los resultados, contamos que partimos de una base de datos o *dataset* que almacena la siguiente información contrastada con los resultados obtenidos:

Tabla 22. Análisis de resultados del sistema de clasificación.

	Nº de manzanas
Contenidas en la <i>GDB</i>	4862
Contenidas en el <i>dataset</i>	2600
Clasificadas en el <i>dataset</i>	1787
De tipo A	710
De tipo B	312
De tipo C	765

Sin clasificar en la <i>GDB</i>	3075
Otras	813

Gracias a esta tabla podemos llegar a la conclusión de que nuestro trabajo reduce una tercera parte el del equipo cartográfico, clasificando 1787 manzanas del *dataset* pendiente. Esto es una estimación positiva, aunque una vez comprobada la base de datos, el porcentaje de acierto podría menor. En la Figura 36 se puede ver el número de manzanas, desglosadas por tipos, que nuestro sistema ha sido capaz de clasificar.

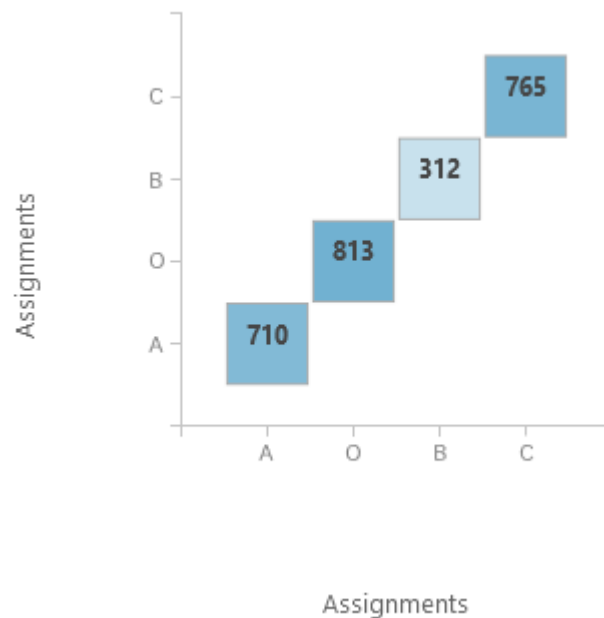


Figura 31. Distribución de la variable clase de manzanas.

Analizando los resultados vemos que el tipo de manzana que más casos tiene es el Tipo C, con 765 casos. Por otro lado, el tipo de manzana que menos casos tiene es el Tipo B, con 312 casos. Podemos ver esta distribución en la Figura 31. Por último, para evaluar la bondad del modelo creado añadimos un nuevo bloque llamado *Evaluate Model*, como podemos ver en la Figura 32.

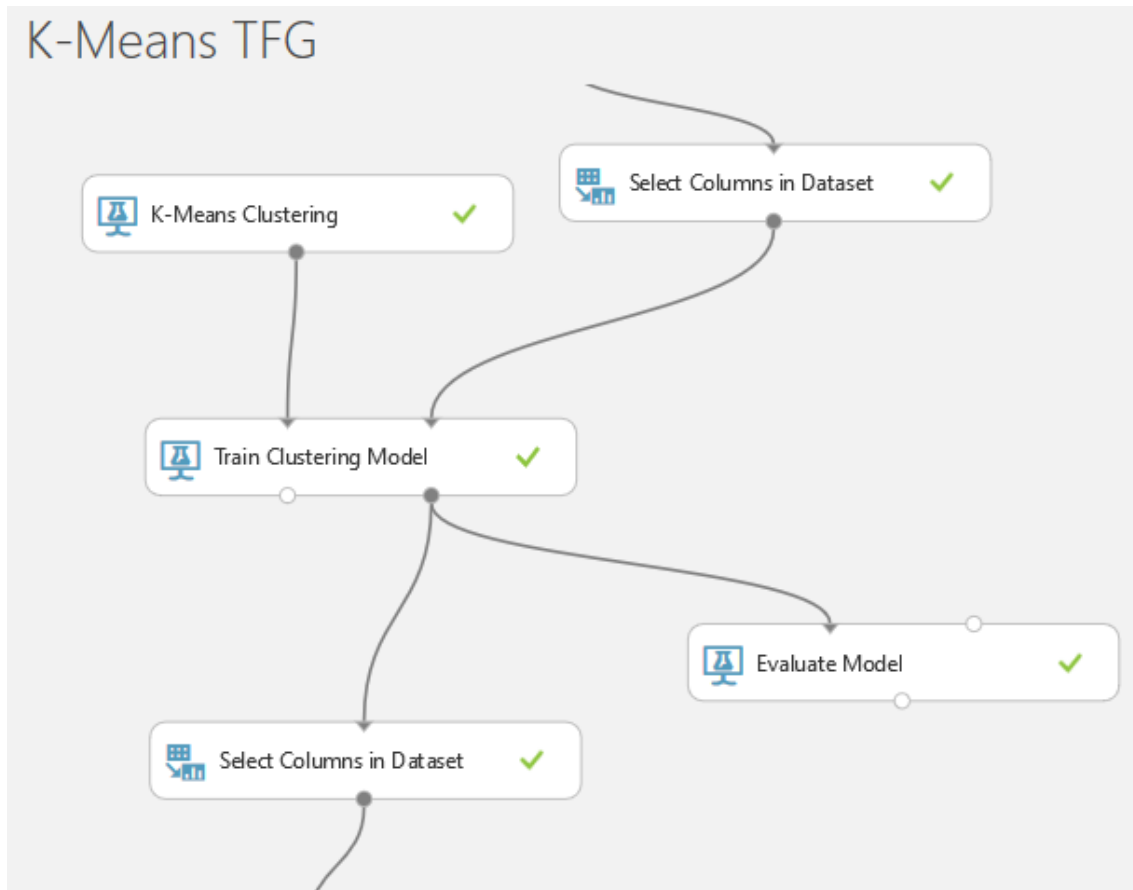


Figura 32. Esquema del experimento tras añadir el módulo de evaluación.

Este nos permite, a partir del dataset de salida que nos da como resultado el entrenamiento del modelo, generar un conjunto de métricas entre las que se encuentran: el número de puntos de datos que se asignaron a cada cluster o grupo, la cantidad de separación entre grupos o clústeres y que tan cohesionados se encuentran los clústeres o grupos con respecto a sus datos.

K-Means TFG > Evaluate Model > Evaluation results

rows

5

columns

5






	Result Description	Average Distance to Cluster Center	Average Distance to Other Center	Number of Points	Maximal Distance To Cluster Center
view as					
	Combined Evaluation	3.233193	9.923924	2600	44.774864
	Evaluation For Cluster A	3.471816	7.510598	710	15.540386
	Evaluation For Cluster O	1.942615	13.11899	813	10.222104
	Evaluation For Cluster B	4.25799	10.984496	312	44.774864
	Evaluation For Cluster C	3.965326	8.343192	765	12.305464

Figura 33. Resultados de evaluación del modelo tratado.

Podemos ver los resultados de la evaluación del modelo en la Figura 33, en la cual podemos analizar el score o puntuación media de los clústeres en la fila *Combined Evaluation*, donde vemos que en la columna la *Distance to Cluster Center* que representa la cercanía de todos los puntos de un grupo al centroide del mismo, nuestro algoritmo cuenta con un 3.233193 de media, siendo el grupo de manzanas sin clasificar el que menos distancia tiene al centro y el grupo de manzanas tipo B el que más. También podemos ver como en la columna *Distance to Other Center*, nuestros clústeres cuentan con valores no demasiado elevados, por lo que deja ver que nuestro clústeres no están bien diferenciados, como efectivamente vemos en la Figura 36. Como último dato de evaluación de nuestro modelo vemos que el valor de la columna *Distance To Cluster Center* no es demasiado alto, contando de media con una puntuación de 44.77, por lo que significa que nuestro clústeres no están demasiado dispersos, cosa que ya fue confirmada previamente en el valor de la distancia de los puntos al centro del cluster, lo cual ayuda en la especialización de los mismos.

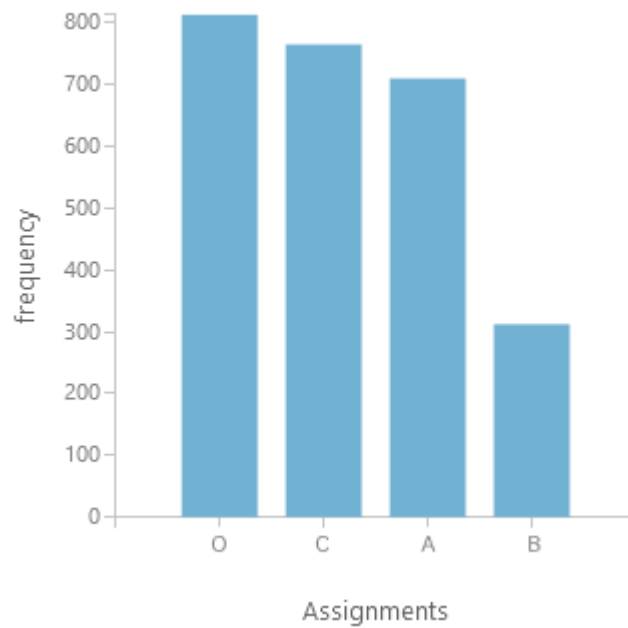


Figura 34. Gráfica de barras de la distribución de la variable clase de manzanas.

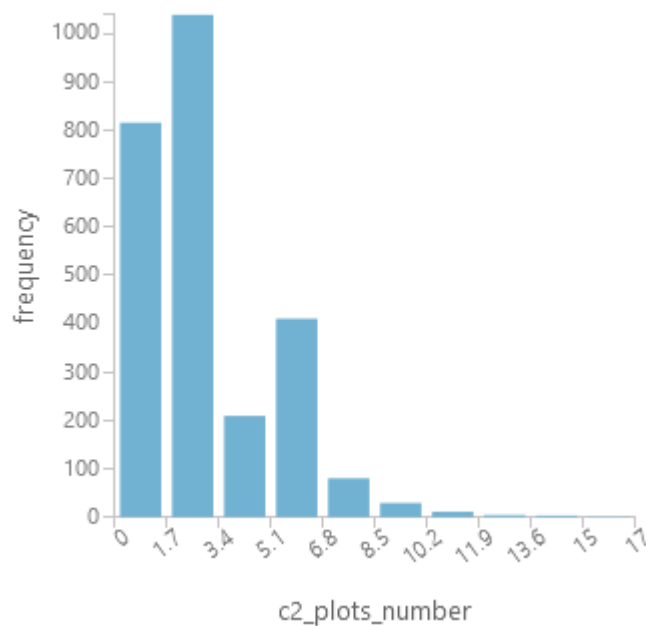


Figura 35. Distribución de la cantidad de lotes en el lado c2 de las manzanas.

Finalmente, podemos ver el diagrama de clústeres creados en la Figura 36. Analizándolo, podemos ver que se crean cuatro *clústeres* autocontenidos unos con otros, en los que se agrupan los diferentes casos recogidos en el dataset. Cada cluster muestra la forma del conjunto dentro del espacio de representación. Esto nos permite valorar que la distinción entre tipos de manzanas no es del todo

precisa, por lo que nuestro algoritmo no ha sido capaz de diferenciar claramente entre unos tipos y otros con exactitud, siendo únicamente los clústeres correspondientes a las manzanas de Tipo A y B las únicas que consiguen abarcar zonas en solitario del espacio de representación de la muestra. Además, la imprecisión de los datos hace que el sistema no sea capaz de diferenciar claramente entre los diferentes tipos de manzanas. Esto es debido a que sus variables principales de clasificación son los lotes con los que cuenta cada manzana en cada uno de sus lados y esta clasificación no es del todo precisa. Como se puede ver en la Figura 35, en la que se muestra la distribución del número de lotes que tienen las manzanas del dataset en el lado c2 (lado corto más al sur).

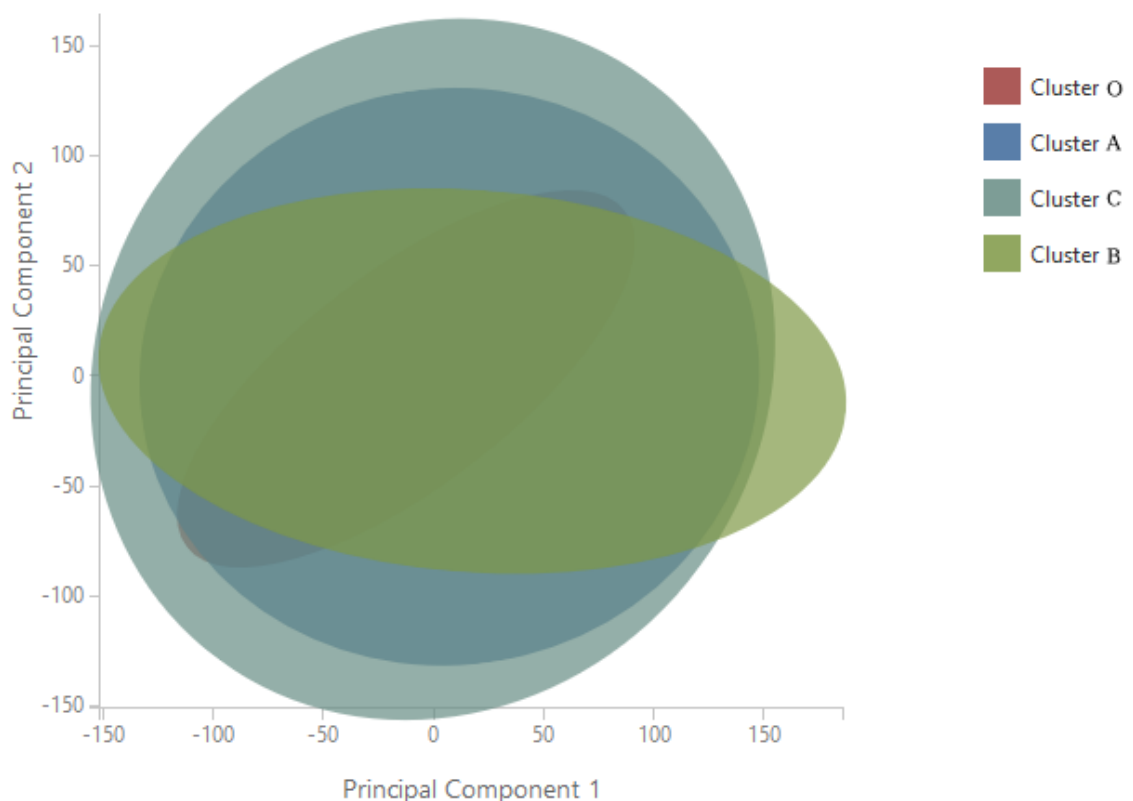


Figura 36. Gráfico que muestra los *clústeres* de manzanas creados.

También la realización de un preprocesado de los datos, excluyendo ciertos patrones conocidos como por ejemplo las manzanas que no cuentan con una morfología apropiada para su análisis, habría mejorado el resultado. Sin embargo, esto podría haber eliminado ciertos patrones interesantes por descubrir, como por ejemplo posibles nuevos tipos de manzanas.

5.8 Conclusiones

En este capítulo se ha creado un sistema de clasificación de elementos de la cartografía base. Para ello, se ha realizado previamente un análisis de las necesidades del sistema y se ha construido una base de conocimiento candidata de ser clasificada por el sistema creado. Además, se ha creado el experimento de clasificación en la plataforma Microsoft Azure Machine Learning Studio, para después realizar un análisis de los resultados obtenidos. El análisis de estos resultados nos hace ver que se ha conseguido reducir un tercio el trabajo del equipo cartográfico, lo que en tiempo del proyecto se traduce en meses.

Tras el trabajo de clasificación de la base de datos de elementos cartográficos, se plantea la necesidad de crear herramientas que digitalicen de manera correcta los lotes mal formados e identificados a partir de los resultados del sistema de clasificación. Para ello se realizan los trabajos documentados en el Anexo I.

Capítulo 6

Conclusiones y trabajos futuros

6.1 Introducción

En este último capítulo se pretende poner un punto final al trabajo realizado, reflexionando sobre las tareas llevadas a cabo de acuerdo a los resultados obtenidos, midiendo el valor de las soluciones, con los objetivos marcados a conseguir al principio de este.

También se da una guía de continuación de este trabajo indicando el camino a seguir para su crecimiento o ampliación, y se critica la forma en que se trabaja, buscando una mejora profesional de cara a futuros trabajos de características similares.

6.2 Conclusiones

Al comienzo de este TFG se definieron, en la Sección 1.2, los objetivos a alcanzar con nuestro trabajo. Por ello ahora recorreremos otra vez esos puntos para valorar con mirada crítica, las soluciones implementadas:

- OBJ 1. Estudio y análisis de los Sistemas de Información Geográfica. Este primer objetivo propuesto se completa con la introducción realizada a los SIG en la Sección 2.2. En ella se pone en contexto este TFG, dando una primera aproximación a estos sistemas, su historia y los elementos que los forman.
- OBJ 2. Estudiar y analizar las técnicas de Machine Learning. En este objetivo se quiere hacer una introducción al Machine Learning, dando una primera definición del término, explicando sus elementos principales, así como sus

técnicas y algoritmos. Esto se realiza en la Sección 2.4, la cual consigue poner al lector en contexto antes de comenzar la lectura de las secciones posteriores en las que se ven muchos de los términos y referencias al Machine Learning que se definen.

- OBJ 3. Estudio de necesidades. Este objetivo se consigue con la realización, en cada capítulo en el cual se necesita, de un análisis de tecnologías candidatas con las que implementar la solución. En ellas podemos encontrar, además, tablas comparativas que nos ayudan a entender el porqué de nuestra decisión.
- OBJ 4. Codificación de la cartografía base. En este objetivo, el cliente necesitaba una numeración que permitiera la identificación de los elementos cartográficos de esta posteriormente en los trabajos en campo. Este objetivo se ha conseguido con la entrega de una cartografía almacenada en formato *GDB*, compuesta por todos y cada uno de los elementos de ella identificados por un atributo entero y único. Dada la complejidad de las soluciones pedidas, se formulan diversos subobjetivos que repasamos a continuación:
 - RF1.1. Asignación de códigos de manzana. Este subobjetivo se consiguió gracias a la realización de un algoritmo de numeración, según los criterios pedidos, en la Sección 4.6, creándose un atributo de capa, entero y único para cada elemento. Sin embargo, el resultado final de numeración no es tan estético como se desearía, debido a que las franjas de numeración horizontales que proponemos, en varias ocasiones, quedan vacías de manzanas que cumplan los criterios de superposición, por lo que el resultado final parece ambiguo a simple vista, aunque es correcto en cálculo. También influye la morfología de los subsectores que contienen a las manzanas, debido a que, como es lógico, la cartografía es caprichosa, no existiendo el caso de numeración perfecto. Esto ha añadido gran complejidad a la casuística que ha tenido que ser contemplada en la solución.
 - RF1.2. Asignación de códigos de esquina. La asignación de códigos de esquinas queda satisfecha en la Sección 4.7, obteniendo como resultado una capa nueva con cada una de las geometrías puntos de las esquinas, con su identificador entero y único que las identifica.

La consecución de este objetivo estuvo marcada por las dificultades que se tuvo con la cartografía base, en la que una mala codificación de las geometrías de las manzanas no permitía obtener los vértices originales de las geometrías. Sin embargo, gracias a la fácil normalización que la plataforma *ArcMap* permitió hacer, se pudo seguir con los trabajos, terminando fácilmente la tarea.

- RF1.3. Asignación de códigos de lote. En este caso, el resultado obtenido de los trabajos de numeración de los lotes de la cartografía, explicados en la Sección 4.8, es un atributo que se suma a la capa de lotes de la cartografía original, en el que se identifica cada geometría con un identificador único y entero. Hubo dificultades iniciales dado que los lotes iniciales con los que se trabajó no se encontraban bien codificados. Esto ralentizó el proceso, que requirió de una codificación inicial de algunas de las geometrías, adaptando su forma a la de las manzanas.
- OBJ 5. Creación de un sistema de clasificación de elementos cartográficos. El último de los objetivos se ve completado en el Capítulo 5, en el que se hace un estudio previo de necesidades y se termina la creación de un sistema inteligente de clasificación de manzanas cartográficas según los criterios establecidos. El resultado es un modelo entrenado listo para clasificar cualquier dataset disponible de manzanas cartográficas que contengan los atributos establecidos. Además, se realizó una clasificación de los elementos con los que trabajamos, recogida en un atributo de clase, entero y discreto, añadido a la capa manzanas.

Haciendo una valoración de los resultados del modelo, podemos ver, en una revisión manual que, aunque los resultados son buenos para los tiempos del proyecto, codificando la mayoría de las manzanas de manera correcta, muchas otras quedan mal codificadas, como se esperaba. Esto se debe a que el modelo no es perfecto, debido al gran ruido que contiene el dataset con el que se trabaja, conteniendo una distribución no uniforme entre los valores de sus variables como podemos ver en la Sección 5.7.

6.3 Trabajos futuros y mejoras

Debido al carácter empresarial de este proyecto, teniendo un resultado potencial y tangible, definido previamente en un pliego de condiciones, queda poco

espacio para continuaciones o mejoras, aunque sí que podemos proponer algunas pautas a seguir en futuros trabajos de la misma índole o temática, de cara a mejorar las soluciones:

- **Creación de *scripts* de autodetección de errores cartográficos:** Esto habría reducido de manera notable la complejidad afrontada a la hora de codificar y numerar la cartografía. Además, habría incrementado la exactitud del sistema de clasificación creado posteriormente.
- **Ejecución de un preprocesamiento de los *datasets* tratados:** Este paso previo a la creación del modelo habría sido interesante de cara a limpiar mucho ruido procedente de la mala codificación de la cartografía. También habría facilitado descubrir ciertas relaciones entre las variables para ser más precisos en la elección de estas para su análisis.
- **Creación de mapas temáticos con *ArcMap*:** El sistema de *ArcGIS* cuenta con potencial para elaborar determinadas estadísticas utilizando mapas, que ayudan a detectar ciertos patrones indetectables a simple vista sobre los datos binarios del *dataset*.

6.4 Valoración personal del trabajo realizado

En este apartado me gustaría hacer un pequeño análisis o valoración global del proyecto, en primera persona, bajo mi punto de vista. Personalmente, estos meses de trabajo para mí han sido de especial enriquecimiento técnico de cara a mi carrera en el mundo de la ingeniería informática. Me han permitido acercarme a tecnologías desconocidas para mí hasta el momento, como son los sistemas de información geográfica, o SIG, en concreto el ya citado en muchas ocasiones, el paquete *ArcGIS* de *ESRI*. Además, me ha permitido conocer el mundo de la cartografía un poquito más, gracias a los trabajos de mis compañeros de proyecto. También he sido participe del ciclo de vida de un proyecto empresarial, desde su creación hasta su entrega, con sus bienaventuranzas y sus problemas.

Haciendo ahora una valoración a nivel de proyecto, podemos decir que hemos reducido el tamaño del conjunto de manzanas que se deben remodelar en una tercera parte. Esto ha ahorrado cerca de un mes de trabajo a los compañeros del equipo de cartografía, los cuales realizan este trabajo una a una, analizando

su morfología a través de métodos de imágenes de satélite. Además, se ha cumplido con creces los requisitos del pliego de condiciones en torno a la numeración de los elementos de la cartografía.

Por último, en el aspecto académico, he sido capaz de poner en práctica no solo todos y cada uno de los conceptos y habilidades que he adquirido durante mi periplo universitario, si no también otros valores que se enseñan en las clases, y en las bibliotecas, como son la capacidad de autodidacta, el aprendizaje acelerado y otras capacidades no establecidas en las competencias de la escuela, como la no desesperación y la búsqueda de soluciones.

6.5 Competencias desarrolladas

Este trabajo ha permitido la adquisición de las diferentes competencias relacionadas con la rama de Informática propia a los estudios cursados. Todas estas competencias son extraídas de la web oficial de la Escuela Superior de Ingeniería Informática (ESIIAB, 2019).

Tabla 23. Competencias desarrolladas en el proyecto

Competencia	Descripción	Justificación
Competencias específicas de la rama de Computación		
[CM1]	Capacidad para tener un conocimiento profundo de los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la informática.	Este trabajo se inicia sobre un pliego de condiciones establecidos en el proyecto por un cliente. Estos son interpretados y puestos en marcha por el equipo técnico de trabajo, obteniendo finalmente un producto entregable.
[CM3]	Capacidad para evaluar la complejidad computacional	Haciendo uso de la descripción dada en

	de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.	[CO6], este trabajo se basa en la reducción de tiempos computacionales y empresariales de cara a la codificación de la cartografía en su primera parte.
[CM4]	Capacidad para conocer los fundamentos, paradigmas y técnicas propias de los sistemas inteligentes y analizar, diseñar y construir sistemas, servicios y aplicaciones informáticas que utilicen dichas técnicas en cualquier ámbito de aplicación.	Nos basamos en la competencia previa [CO15]. Esto queda demostrado en la segunda parte de nuestro trabajo en la que aprovechamos las ventajas de la inteligencia computacional para reducir los tiempos de remodelación de la cartografía.

Anexo I. Caja de herramientas ArcGIS para la digitalización de lotes

Tras la creación del sistema de clasificación de elementos cartográficos y la obtención de una base de conocimiento organizada por etiquetas que denotan el tipo de manzana, el proyecto requiere la creación de una caja de herramientas, o *toolbox* de ArcGIS. Esta ha de permitir dividir estas manzanas en sus respectivos lotes dependiendo del tipo de manzana con el que nos encontremos como se ve en el capítulo 4 de esta memoria.

Las *toolbox* o cajas de herramientas de ArcGIS para ArcMap son un conjunto de programas, scripts, propios de la plataforma, que te permiten ejecutar una acción o tarea automatizada de geoprocésamiento a partir de una abstracción de los datos de entrada. En esta ocasión las herramientas son propias, personalizadas, por lo que los datos de entrada están muy sesgados por el desarrollo, no permitiendo una reutilización de la herramienta para otros usos.

El programa ArcMap ya cuenta con una colección de *toolboxes* propias, englobadas en las siguientes categorías, propiamente definidas en su documentación oficial (ESRI, 2019b).

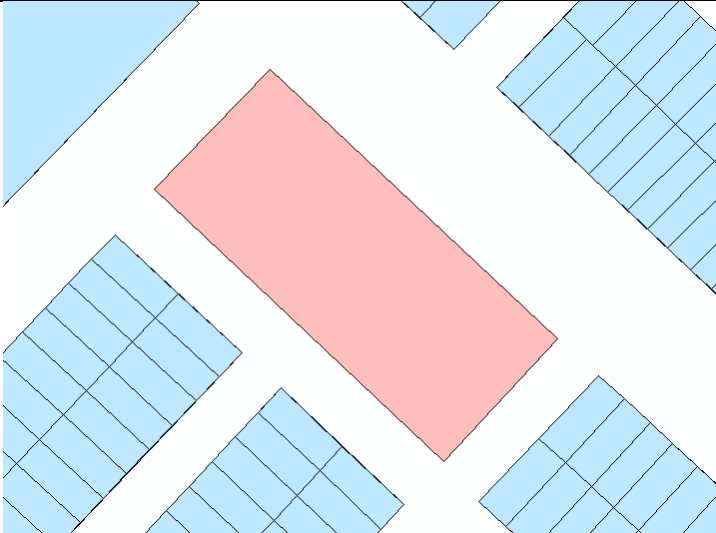
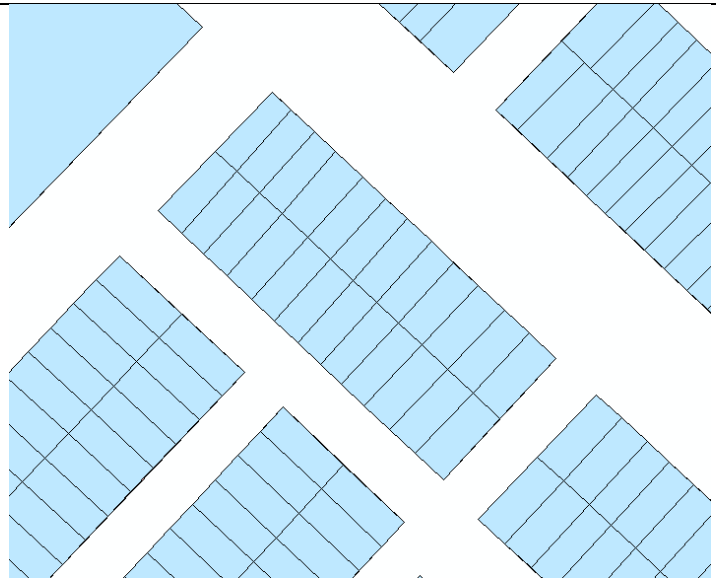
Nuestra tarea ha sido crear una nueva *toolbox* que permita la remodelación de los lotes incluidos en los diferentes tipos de manzanas especificados en Tabla 11.

Para ello se creó un fichero *Python Toolbox* con extensión *pyt*, y siguiendo la plantilla que nos proporciona *ArcMap*, creamos cuatro herramientas dentro de la *toolbox* Remodelar Lotes:

I.I. Manzanas Tipo 1

Tabla 24. Descripción de la herramienta de remodelación de lotes de manzanas de tipo 1.

	Parámetro	Tipo	Descripción
Parámetros de entrada	<i>Path</i> de la gdb sobre la que se ejecuta el algoritmo	<i>String</i>	Ruta donde se ubica la gdb que contiene las capas de manzanas y lotes a analizar.
	Número de lotes en los que dividir la manzana en L1 (a lo largo)	<i>Integer</i>	Parcelas catastrales en las que dividir la manzana en el lado L1 (Largo 1 o lado largo más al sur).
	Nombre de la capa de manzanas a analizar	<i>String</i>	Nombre identificador de la capa dentro de la gdb para el soporte de distintas bases de datos.
	Nombre de la capa de lotes a analizar	<i>String</i>	Nombre identificador de la capa dentro de la gdb para el soporte de distintas bases de datos.
Funcionamiento	<ul style="list-style-type: none"> – Se limpia la manzana de lotes. – Se selecciona la manzana con la herramienta “<i>Select Features</i>”. – Se ejecuta la herramienta “<i>Manzanas Tipo 1</i>” ajustando los parámetros deseados. – Se pulsa aceptar y se refresca la vista actual, por ejemplo, moviendo la posición de la navegación en cualquier dirección con la herramienta “<i>Pan</i>”. 		

Algoritmia	<p>Se obtienen los vértices de la geometría y se forma un conjunto con los lados de manzana. Se ordenan con orden a su longitud y se calculan en ellos los puntos por los que se divide la geometría de la manzana. Una vez dividida obtenemos cada uno de los lotes correspondientes al resultado.</p>
Experimento	
 <p>Figura 37. Vista exploratoria previa a la ejecución de la tool para manzanas tipo 1.</p>	
 <p>Figura 38. Vista exploratoria tras la ejecución de la tool para manzanas tipo 1.</p>	

I.II. Manzanas Tipo 2

Tabla 25. Descripción de la herramienta de remodelación de lotes de manzanas de tipo 2.

Parámetros de entrada	Parámetro	Tipo	Descripción
	<i>Path</i> de la gdb sobre la que se ejecuta el algoritmo	<i>String</i>	Ruta donde se ubica la gdb que contiene las capas de manzanas y lotes a analizar.
	Número de lotes en los que dividir la manzana en L1 (a lo largo)	<i>Integer</i>	Parcelas catastrales en las que dividir la manzana en el lado L1 (Largo 1 o lado largo más al sur).
	Número de lotes en los que dividir la manzana C1 (a lo ancho)	<i>Integer</i>	Parcelas catastrales en las que dividir la manzana en el lado C1 (Corto 1 o lado corto más al sur).
	Número de lotes en los que dividir la manzana en C2 (a lo ancho)	<i>Integer</i>	Parcelas catastrales en las que dividir la manzana en el lado C2 (Corto 2 o lado corto más al norte).
	Nombre de la capa de	<i>String</i>	Nombre identificativo de la capa dentro de la

	manzanas a analizar		gdb para el soporte de distintas bases de datos.
	Nombre de la capa de lotes a analizar	String	Nombre identificativo de la capa dentro de la gdb para el soporte de distintas bases de datos.
Funcionamiento	<ul style="list-style-type: none">– Se limpia la manzana de lotes.– Se selecciona la manzana con la herramienta “Select Features”.– Se ejecuta la herramienta “Manzanas Tipo 2” ajustando los parámetros deseados.– Se pulsa aceptar y se refresca la vista actual, por ejemplo, moviendo la posición de la navegación en cualquier dirección con la herramienta “Pan”.		
Algoritmia	Se obtienen los vértices de la geometría y se forma un conjunto con los lados de manzana. Se ordenan con orden a su longitud y se calculan en ellos los puntos por los que se divide la geometría de la manzana. Una vez dividida obtenemos cada uno de los lotes correspondientes al resultado. La diferencia respecto al script de remodelación de manzanas tipo 1 viene al calcular los puntos para dividir la manzana. Aquí se tiene en cuenta la reserva del espacio suficiente para contemplar los lotes laterales a codificar.		
Experimento			

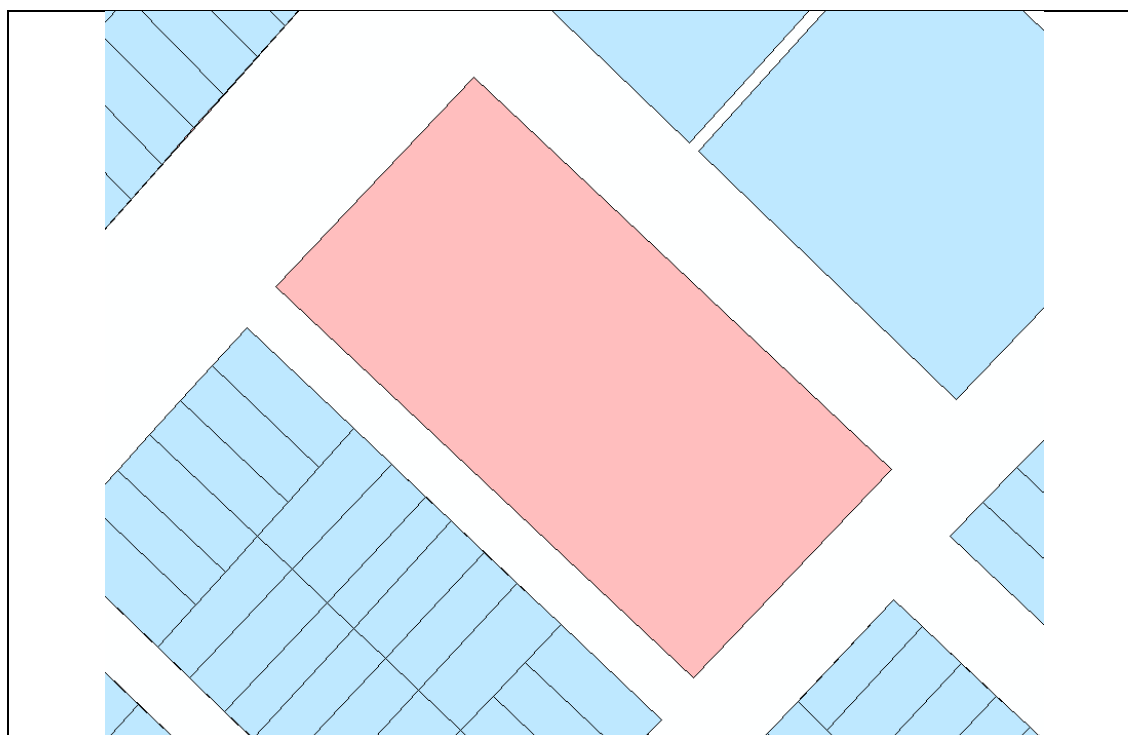


Figura 39. Vista exploratoria previa a la ejecución de la tool para manzanas tipo 2.

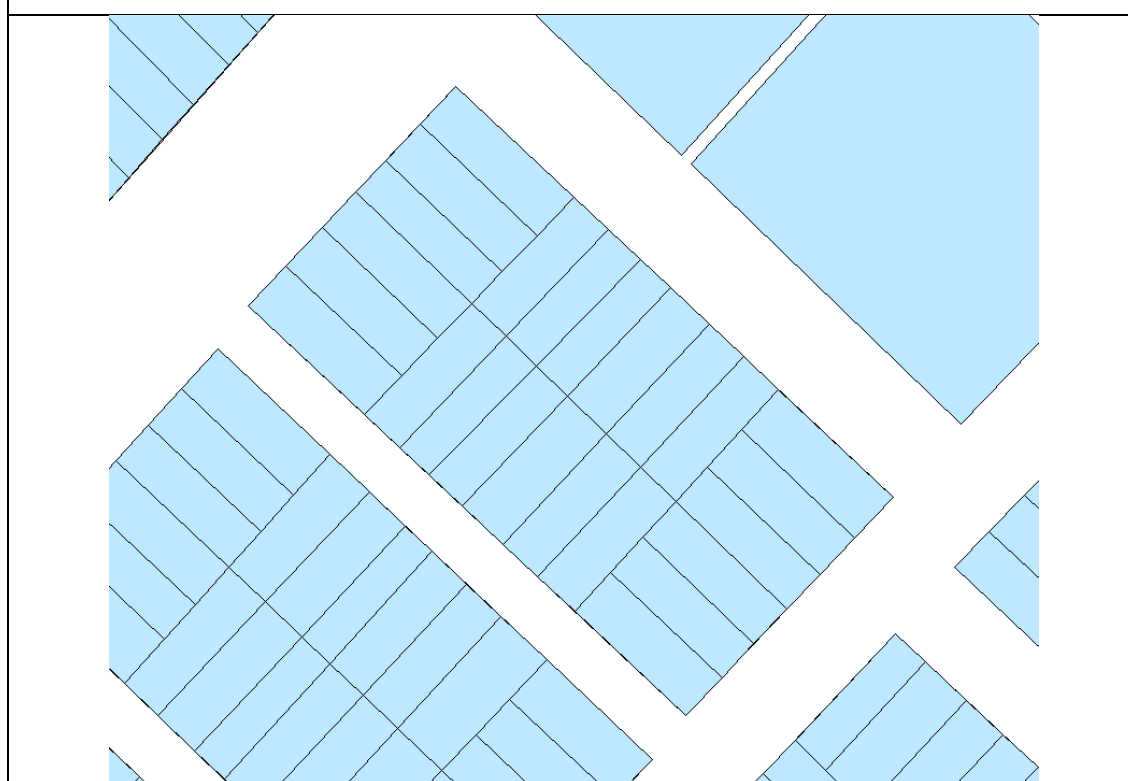


Figura 40. Vista exploratoria tras la ejecución de la tool para manzanas tipo 2.

I.III. Manzanas Tipo 3_A

Tabla 26. Descripción de la herramienta de remodelación de lotes de manzanas de tipo 3_A.

Parámetros de entrada	Parámetro	Tipo	Descripción
	<i>Path</i> de la gdb sobre la que se ejecuta el algoritmo	<i>String</i>	Ruta donde se ubica la gdb que contiene las capas de manzanas y lotes a analizar.
	Número de lotes en los que dividir la manzana en L1 (a lo largo)	<i>Integer</i>	Parcelas catastrales en las que dividir la manzana en el lado L1 (Largo 1 o lado largo más al sur).
	Número de lotes en los que dividir la manzana C1 (a lo ancho)	<i>Integer</i>	Parcelas catastrales en las que dividir la manzana en el lado C1 (Corto 1 o lado corto más al sur).
	Nombre de la capa de manzanas a analizar	<i>String</i>	Nombre identificador de la capa dentro de la gdb para el soporte de distintas bases de datos.
	Nombre de la capa de lotes a analizar	<i>String</i>	Nombre identificador de la capa dentro de la gdb para el soporte

			de distintas bases de datos.
Funcionamiento	<ul style="list-style-type: none">– Se limpia la manzana de lotes.– Se selecciona la manzana con la herramienta “<i>Select Features</i>”.– Se ejecuta la herramienta “<i>Manzanas Tipo 3_A</i>” ajustando los parámetros deseados.– Se pulsa aceptar y se refresca la vista actual, por ejemplo, moviendo la posición de la navegación en cualquier dirección con la herramienta “<i>Pan</i>”.		
Algoritmia	<p>Se obtienen los vértices de la geometría y se forma un conjunto con los lados de manzana. Se ordenan con orden a su longitud y se calculan en ellos los puntos por los que se divide la geometría de la manzana. Una vez dividida obtenemos cada uno de los lotes correspondientes al resultado. Al igual que la herramienta anterior, lo que diferencia a la <i>tool</i> de remodelación de lotes en manzanas de tipo 3_A es la reserva del espacio en uno de los laterales que alberga las parcelas con salida exterior a la calle en el extremo de la manzana.</p>		
Experimento			

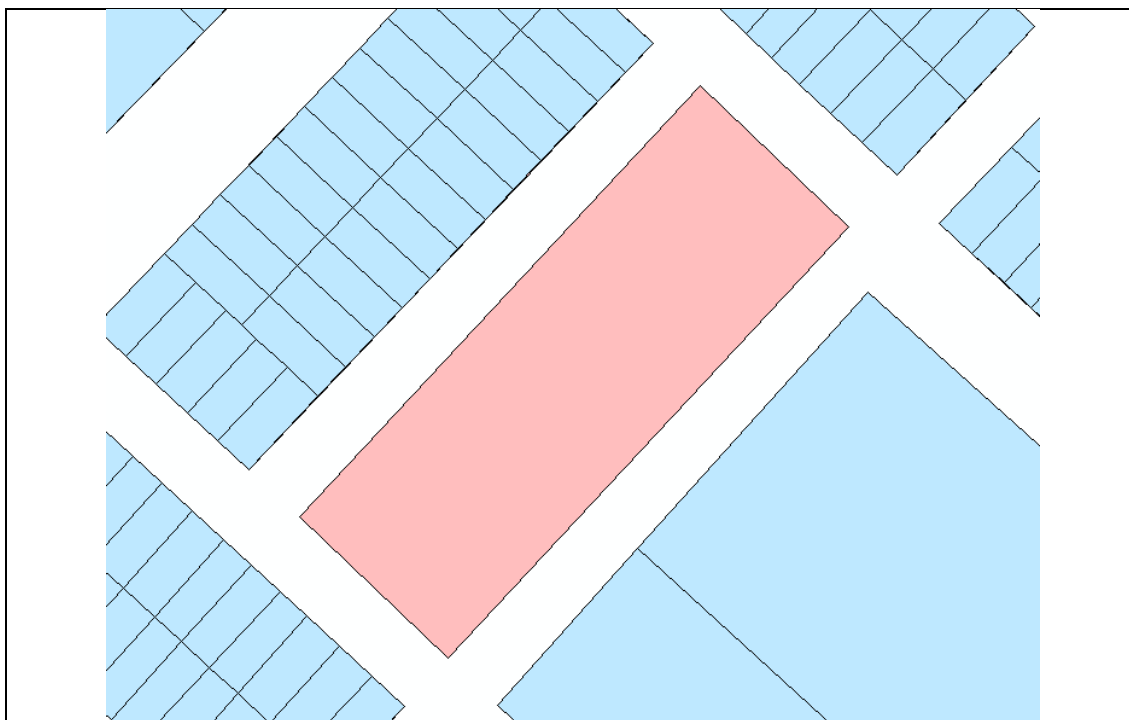


Figura 41. Vista exploratoria previa a la ejecución de la tool para manzanas tipo 3_A.

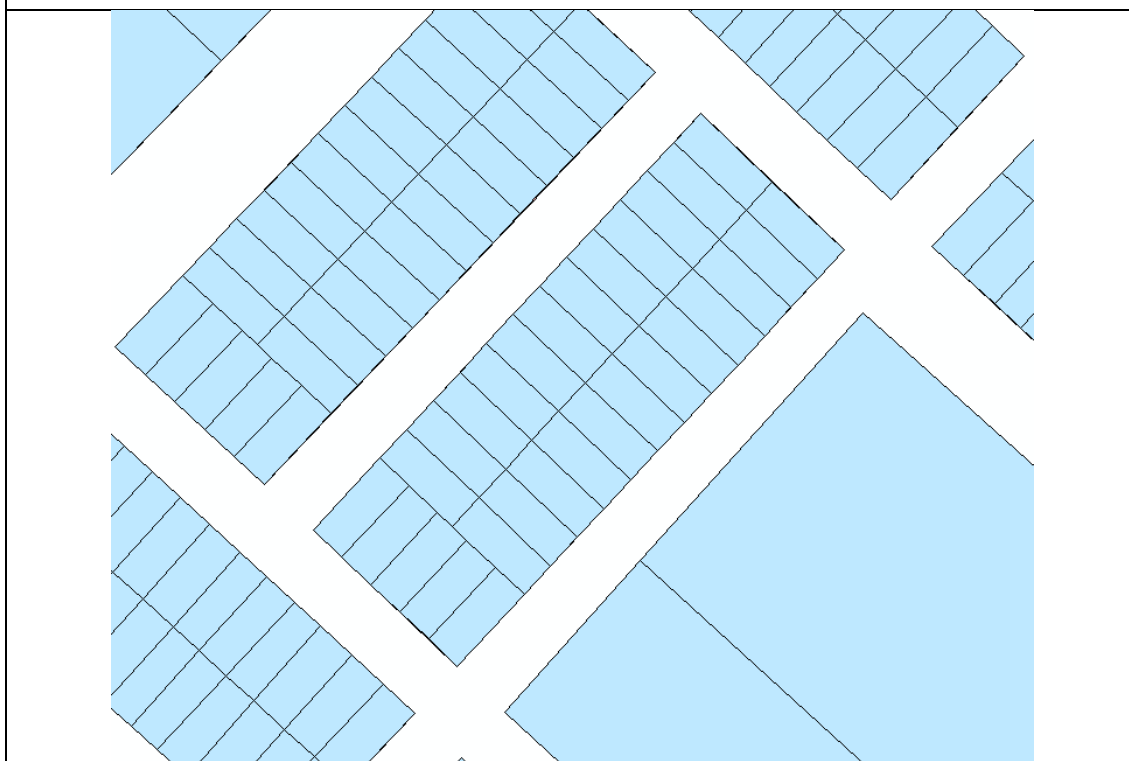


Figura 42. Vista exploratoria tras la ejecución de la tool para manzanas tipo 3_A.

I.IV. Manzanas Tipo 3_B

Tabla 27. Descripción de la herramienta de remodelación de lotes de manzanas de tipo 3_B.

Parámetros de entrada	Parámetro	Tipo	Descripción
	<i>Path</i> de la gdb sobre la que se ejecuta el algoritmo	<i>String</i>	Ruta donde se ubica la gdb que contiene las capas de manzanas y lotes a analizar.
	Número de lotes en los que dividir la manzana en L1 (a lo largo)	<i>Integer</i>	Parcelas catastrales en las que dividir la manzana en el lado L1 (Largo 1 o lado largo más al sur).
	Número de lotes en los que dividir la manzana en C1 (a lo ancho)	<i>Integer</i>	Parcelas catastrales en las que dividir la manzana en el lado C1 (Corto 1 o lado corto más al sur).
	Nombre de la capa de manzanas a analizar	<i>String</i>	Nombre identificador de la capa dentro de la gdb para el soporte de distintas bases de datos.
	Nombre de la capa de lotes a analizar	<i>String</i>	Nombre identificador de la capa dentro de la

			gdb para el soporte de distintas bases de datos.
Funcionamiento	<ul style="list-style-type: none">– Se limpia la manzana de lotes.– Se selecciona la manzana con la herramienta “Select Features”.– Se ejecuta la herramienta “Manzanas Tipo 3_B” ajustando los parámetros deseados.– Se pulsa aceptar y se refresca la vista actual, por ejemplo, moviendo la posición de la navegación en cualquier dirección con la herramienta “Pan”.		
Algoritmia	<p>Se obtienen los vértices de la geometría y se forma un conjunto con los lados de manzana. Se ordenan con orden a su longitud y se calculan en ellos los puntos por los que se divide la geometría de la manzana. Una vez dividida obtenemos cada uno de los lotes correspondientes al resultado.</p> <p>Algunos problemas derivados del proyecto obligan a crear esta herramienta idéntica a la anterior con la salvedad de que se tiene en cuenta la orientación sur norte del extremo de la manzana que cuenta con parcelas catastrales con salida al exterior.</p>		
Experimento			

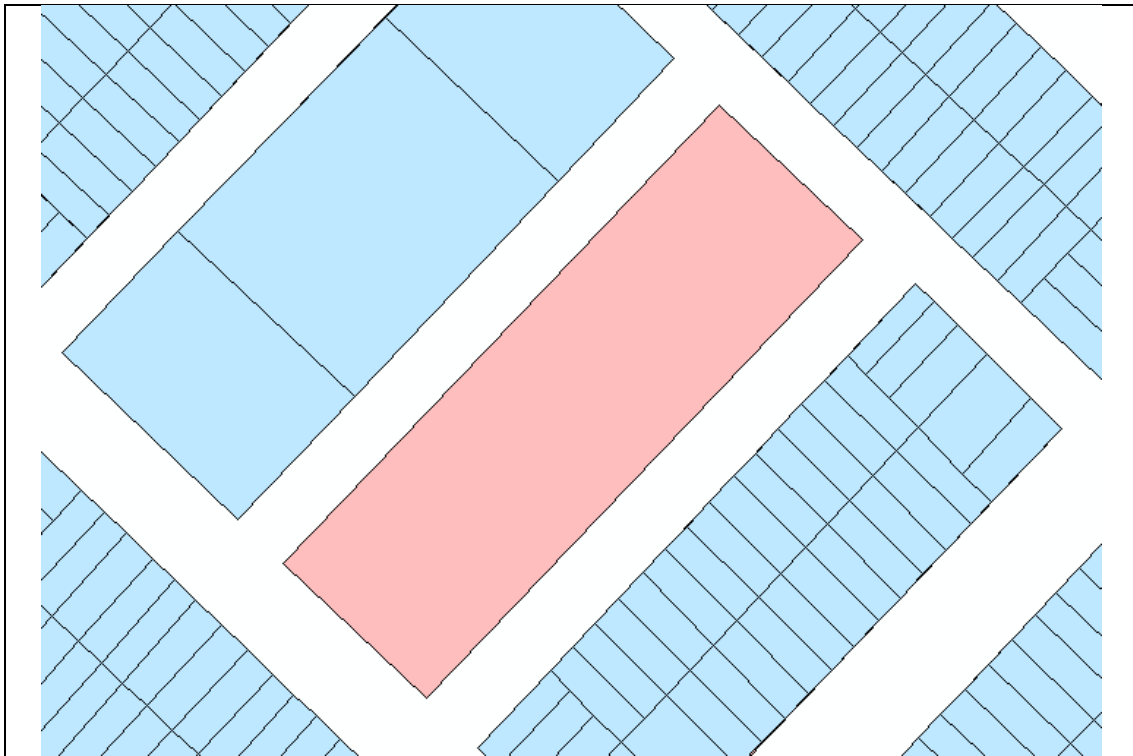


Figura 43. Vista exploratoria previa a la ejecución de la tool para manzanas tipo 3_B.

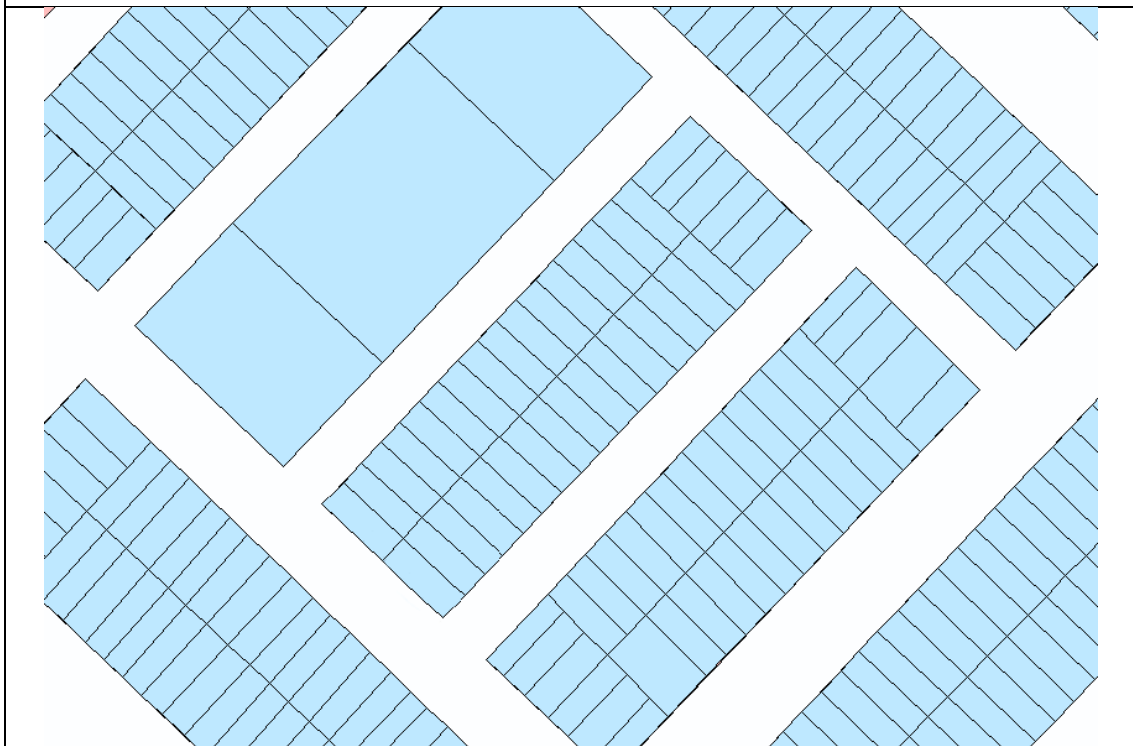


Figura 44. Vista exploratoria tras la ejecución de la tool para manzanas tipo 3_B.

Anexo II. Glosario de términos

ArcGIS: Conjunto de software perteneciente a la familia de los Sistemas de Información Geográfica (SIG), creado y mantenido por la marca ESRI, es usado para la captura, tratamiento, análisis y edición de información geográfica.

ArcMap: Es uno de los programas que forman parte de la familia de componentes ArcGIS de Esri. Su funcionalidad principal es permitir editar, ver, crear y analizar datos espaciales. Esto es posible gracias a su potente visor por capas y su conjunto de herramientas para el manejo de datos geoespaciales.

Geodatabase o GDB: Estructura de almacenamiento nativa de ArcGIS para el guardado de conjuntos de datasets de información geoespacial, bien contenida en un sistema de ficheros (carpetas) básico o en un sistema gestor de bases de datos.

Integer: Tipo de datos que representa los números enteros en los lenguajes de programación.

Path: Referencia a un archivo o directorio determinado en un sistema operativo concreto dentro de la jerarquía de un sistema de ficheros.

RGB: Modelo de color representado por la addición de los tres colores primarios, del que toma su nombre en inglés (rojo, verde y azul).

Script: Conjunto de líneas de programación escritas en alguno de los lenguajes conocidos, cuyo fin es ejecutar diferentes procesos para conseguir resolver una tarea determinada.

String: Tipo de datos que representa una cadena de caracteres en un lenguaje de programación.

Toolbox: Conjunto de procesos instrumento que sirven para un fin similar dentro del procesamiento de información geográfica. Pueden ser propias o del sistema, al tratarse de las cajas de herramientas que vienen por defecto en el sistema ArcGIS.

Bibliografía

- ¿Qué es ArcPy?—Help | ArcGIS for Desktop. (n.d.). Retrieved June 5, 2019, from <http://desktop.arcgis.com/es/arcmap/10.3/analyze/arcpy/what-is-arcpy-.htm>
- Almoguera Millán, C., Tarquis Alfonso, A. M., & Hernández Díaz-Ambrona, C. G. (2009). El nuevo SIGPAC DEHESA, una herramienta fundamental en la aplicación de la nueva normativa de cerdo ibérico. *Ganadería*, 7(58), 34–37.
- Arreguin, F. (2015). El uso eficiente del agua y la tecnología. *Tecnología y Ciencias Del Agua*, 12(1), 91–98.
- Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 1027–1035).
- Awad, W. A., & ELseuofi, S. M. (2011). Machine learning methods for spam e-mail classification. *International Journal of Computer Science & Information Technology (IJCSIT)*, 3(1), 173–184.
- Barga, R., Fontama, V., & Tok, W. H. (2015). Introducing microsoft azure machine learning. In *Predictive Analytics with Microsoft Azure Machine Learning* (pp. 21–43). Springer.
- Cambronero, C. G., & Moreno, I. G. (2006). Algoritmos de aprendizaje: knn & kmeans. *Inteligencia En Redes de Comunicación, Universidad Carlos III de Madrid*.
- Correia, P. (2002). *Guía práctica del GPS*. Marcombo.
- Daniel, L., Loree, P., & Whitener, A. (2002). *Inside MapInfo professional: the friendly user guide to MapInfo professional*. Cengage Learning.
- de Fomento, C. (Ed.). (2009). *Sistemas de localización e información geográfica*. [Valladolid]: Consejería de Fomento. Retrieved from <http://bibliotecadigital.jcyl.es/i18n/consulta/registro.cmd?id=18571>
- Díaz, L. T. (2013). Intimidación y extimidad en las redes sociales. Las demarcaciones éticas de Facebook. *Comunicar: Revista Científica Iberoamericana de Comunicación y Educación*, (41), 205–213.

- Dlib. (2019a). dlib C++ Library. Retrieved December 1, 2019, from <http://dlib.net/>
- Dlib. (2019b). dlib C++ Library - Machine Learning. Retrieved December 1, 2019, from <http://dlib.net/ml.html>
- Eptisa. (2019). Que hacemos | Eptisa Servicios de Ingeniería ¿Quiénes somos? Retrieved October 24, 2019, from <http://www.eptisa.com/es/empresa/que-hacemos/>
- ESIAB. (2019). Escuela Superior de Ingeniería Informática. Retrieved November 16, 2019, from <https://esiiab.uclm.es/grado/objetivos.php?curso=2019-20&idmenup=objetivosgrado>
- ESRI. (n.d.). Leer geometrías—ArcPy Get Started | ArcGIS Desktop. Retrieved November 30, 2019, from <https://pro.arcgis.com/es/pro-app/arcpy/get-started/reading-geometries.htm>
- ESRI. (2019a). ¿Qué es una geodatabase?—Ayuda | ArcGIS for Desktop. Retrieved November 20, 2019, from <http://desktop.arcgis.com/es/arcmap/10.3/manage-data/gdb-architecture/what-is-a-geodatabase.htm>
- ESRI. (2019b). Un paseo introductorio por las referencias de las herramientas de geoprocésamiento—Ayuda | ArcGIS for Desktop. Retrieved November 4, 2019, from <https://desktop.arcgis.com/es/desktop/latest/main/tools/a-quick-tour-of-geoprocessing-tool-references.htm>
- Faggella, D. (2019). What is Machine Learning. 2019. URL: <https://Emerj.Com/Ai-Glossary-Terms/What-Is-Machinelearning>.
- Fritz, S., & Wexler, H. (1960). Cloud pictures from satellite TIROS I. *Monthly Weather Review*, 88(3), 79–87.
- Kompare, B., Bratko, I., Steinman, F., & Džeroski, S. (1994). Using machine learning techniques in the construction of models I. Introduction. *Ecological Modelling*, 75, 617–628.
- Konecny, G. (2004). Small satellites--A tool for Earth observation? In *XXth ISPRS Congress, Commission* (Vol. 4, pp. 12–23).
- Kuang, H., & Xiao, S. (2006). Research of GE Smallworld GIS based outage analysis system. *Advanced Technology of Electrical Engineering and Energy*, 25(4), 71.
- Lauzon, F. Q. (2012). An introduction to deep learning. In *2012 11th International*

- Conference on Information Science, Signal Processing and their Applications (ISSPA)* (pp. 1438–1439).
- Lori Lewis. (2019). 2019: This Is What Happens In An Internet Minute by Lori Lewis. Retrieved November 26, 2019, from <https://www.allaccess.com/merge/archive/29580/2019-this-is-what-happens-in-an-internet-minute>
- Maguire, D. J. (1991). An overview and definition of GIS. *Geographical Information Systems: Principles and Applications*, 1, 9–20.
- Masser, I. (2005). *GIS worlds: creating spatial data infrastructures* (Vol. 338). Esri Press Redlands, CA.
- McHarg, I. L., & Mumford, L. (1969). *Design with nature*. American Museum of Natural History New York.
- Medina, J., Cabeza, E. M. E., & Peña, J. L. C. (2013). Asistentes virtuales en plataformas 3.0. *IE Comunicaciones: Revista Iberoamericana de Informática Educativa*, (18), 41–49.
- Microsoft. (2019a). Hoja de características de los algoritmos de Machine Learning - Azure Machine Learning | Microsoft Docs. Retrieved December 1, 2019, from <https://docs.microsoft.com/es-es/azure/machine-learning/algorithm-cheat-sheet>
- Microsoft. (2019b). Microsoft Azure Machine Learning Studio. Retrieved December 1, 2019, from <https://studio.azureml.net/>
- Molina Ríos, J. R., Loja Mora, N. M., Zea Ordóñez, M. P., & Loaiza Sojos, E. L. (2016). Evaluación de los Frameworks en el Desarrollo de Aplicaciones Web con Python. *Revista Latinoamericana de Ingeniería de Software*, 4(4), 201. <https://doi.org/10.18294/relais.2016.201-207>
- Olmedo, M. T. C. (2006). Sistemas y análisis de la información geográfica. Manual de autoaprendizaje con ArcGIS. *Cuadernos Geográficos*, 39, 231–233.
- Osorio Campillo, H., & Sánchez, E. R. (2011). La cartografía como medio investigativo y pedagógico. *Dearq. Revista de Arquitectura*, (9), 30–47.
- Paul Crickard, Eric van Rees, S. T. (2018). *Mastering Geospatial Analysis with Python*.
- Perlich, C., Dalessandro, B., Raeder, T., Stitelman, O., & Provost, F. (2014). Machine learning for targeted display advertising: Transfer learning in action.

- Machine Learning*, 95(1), 103–127.
- Portilla Jiménez, J. J. (2018). *Análisis y diseño de un sistema de reconocimiento facial aplicando machine learning para detectar e identificar intrusos*. Universidad de Guayaquil. Facultad de Ciencias Matemáticas y Físicas~....
- PostGIS. (2019). PostGIS — Spatial and Geographic Objects for PostgreSQL. Retrieved December 1, 2019, from <https://postgis.net/>
- PostgreSQL. (2019). PostgreSQL: The world's most advanced open source database. Retrieved December 1, 2019, from <https://www.postgresql.org/>
- Python. (n.d.). time — Time access and conversions — Python 3.8.0 documentation. Retrieved November 30, 2019, from <https://docs.python.org/3/library/time.html>
- Santiago, I. (2005). Fundamentos de ArcGis. *Versión ArcView*, 9.
- Scikit learn. (2019a). Choosing the right estimator — scikit-learn 0.21.3 documentation. Retrieved December 1, 2019, from https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html
- Scikit learn. (2019b). scikit-learn: machine learning in Python — scikit-learn 0.21.3 documentation. Retrieved December 1, 2019, from <https://scikit-learn.org/stable/>
- Siabato, W. (2018). Sobre la evolución de la información geográfica: las bodas de oro de los SIG. *Cuadernos de Geografía: Revista Colombiana de Geografía*, 27(1), 1–9.
- Snow, J. (1855). *On the mode of communication of cholera*. John Churchill.
- Soderlund, C., Nelson, W., Shoemaker, A., & Paterson, A. (2006). SyMAP: A system for discovering and viewing syntenic regions of FPC maps. *Genome Research*, 16(9), 1159–1168.
- Sutton, R. S., Barto, A. G., & others. (1998). *Introduction to reinforcement learning* (Vol. 2). MIT press Cambridge.
- Tiwari, A. K. (2017). Introduction to machine learning. *Ubiquitous Machine Learning and Its Applications*. <https://doi.org/10.4018/978-1-5225-2545-5.ch001>
- Toms, S. (2015). *ArcPy and ArcGIS--Geospatial Analysis with Python*. Packt Publishing Ltd.

Unión espacial—Ayuda | ArcGIS for Desktop. (n.d.). Retrieved June 7, 2019, from <http://desktop.arcgis.com/es/arcmap/10.3/tools/analysis-toolbox/spatial-join.htm>

Víctor Olaya. (2006). *Sistemas de Información Geográfica*. Retrieved from <https://volaya.github.io/libro-sig/chapters/Intro.html>

Contenido del CD

El CD que acompaña este TFG contiene los siguientes recursos aquí desglosados:

- Memoria del trabajo fin de grado. **Formato:** *PDF*
- Carpeta con todos los *scripts* de codificación de la cartografía base explicados en el Capítulo Codificación de la cartografía base de esta memoria.
- Carpeta con todos los *scripts* automáticos para la generación de la base de datos con la que reproducir el clasificador de elementos cartográficos creado en el Capítulo Sistema de clasificación de elementos cartográficos de esta memoria.
- Carpeta con la *toolbox* o caja de herramientas de *ArcMap* creada y documentada en el anexo Caja de herramientas ArcGIS para la digitalización de lotes.