UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 3

ALGORITMO DE POST-PROCESAMIENTO DE ESTRATOS PARA LA OBTENCIÓN DE UN DIAGNÓSTICO SOBRE FACTORES DE RIESGO

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor(es):

Dalber I. Morán González

Tutor(es):

Ing. Yadian Guillermo Pérez Betancourt

Ing. Roger Godofredo Rivero Morales

Ing. Liset González Polanco

La Habana, junio de 2018

"Año 60 de la Revolución"

DECLARACIÓN DE AUTORÍA

Declaro ser el autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

_____

Nombre autor

Autor

_____ _____

Tutor 1 **Tutor 2**

Tutor Tutor

DEDICATORIA

A mis padres.

A mis hermanos.

A mis abuelos.

A mis tíos y primos.

En especial a esos que ya no están, pero que formaron parte de mi formación.

RESUMEN

En la actualidad, los Sistemas de Información Geográfica (SIG) son herramientas muy utilizadas en la resolución de los problemas en diversas ramas ya que permiten la captura, administración, manipulación, análisis y presentación de datos georreferenciados. El uso de los SIG en la rama de la salud ha aumentado, fundamentalmente en temas referentes a estratificación de territorios y la identificación de zonas de riesgo. Sin embargo en la literatura consultada se limita en gran medida a la visualización de los estratos como parte del proceso de estratificación.

El objetivo del presente trabajo es desarrollar una propuesta para el post-procesamiento en la estratificación de territorios basada en indicadores de salud. Para ello se realizó un análisis crítico de las herramientas existentes que soportan los procesos de estratificación de territorios. Se definió la arquitectura y los principales patrones de diseño utilizados. Se realizaron las pruebas de software definidas en la metodología de desarrollo seleccionada por el grupo de investigación que implementó la solución en la cual se basa este trabajo. Se aplicó un caso de estudio para valorar los resultados de la solución propuesta.

Se logró como resultado un sistema que permite obtener un diagnóstico sobre posibles factores de riesgo; la solución tiene su base en la estratificación de territorios y contribuye a la identificación de riesgos de salud de los territorios cubanos y a la toma de decisiones en las entidades de salud.

Palabras claves: arquitectura, cartografía de riesgos, estratificación, patrones de diseño, pruebas de software, sistemas de información geográfica, riesgo de salud, salud, post-procesamiento.

ÍNDICE

ÍNDICE

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

INTRODUCCIÓN

En la actualidad, el progresivo avance de las tecnologías computacionales y la necesidad del manejo de información periódicamente en aumento, conlleva a un mayor uso de los sistemas informáticos en diversas actividades de la sociedad. Los Sistemas de Información Geográfica (SIG) son una herramienta informática popular y de gran impacto en los últimos tiempos, teniendo gran impacto en ramas como la agricultura, la meteorología, el turismo y la medicina (VÍCTOR 2011; Shi y Kwan 2015; Yasobant et al. 2015).

El desarrollo de los SIG y su aplicación en diferentes áreas ha brindado la posibilidad de analizar grandes volúmenes de datos espaciales (Tanser y Le Sueur 2015 ; Taha 2016, 2016). El uso de los SIG en la rama de la salud ha cobrado cada día mayor utilidad, su empleo contribuye al fortalecimiento de la capacidad de análisis en materia de salud pública y epidemiológica; brindando información de utilidad a la hora de la toma de decisiones (VÍCTOR 2011; SCHEFER-WENZL et al. 2013; Pérez Betancourt 2018). Por otra parte, facilitan la identificación de la ubicación geográfica de establecimientos de salud y grupos de población que presentan mayor riesgo de enfermar o de morir prematuramente y que, por tanto, requieren de mayor atención preventiva, curativa o de promoción de la salud (Kao et al. 2017).

Para el análisis de las diferentes situaciones de salud, se hace necesario conocer con el mayor grado de detalle posible; las características de cada una de las unidades territoriales, así como, sus grupos poblacionales; a partir, de diferentes indicadores, que pueden ser: demográficos, socio-económicos y ambientales, solo por mencionar algunos. Todos estos elementos tienen un impacto determinante en la caracterización de un territorio y constituyen la base en el establecimiento de la estratificación territorial (Cox et al. 2014; Martín y Barros 2015).

La estratificación territorial es un proceso que permite dimensionar espacialmente los eventos a través de un proceso de agregación y desagregación de los territorios a evaluar, a partir de variables seleccionadas para dichos territorios que permitan agregaciones (por homologías de las características) o desagregaciones (por heterogeneidades de estas) (Batista Moliner et al. 2001; Pérez Betancourt, González Polanco y Febles Rodríguez 2017; Pérez Betancourt 2018). Luego de esta etapa se debe realizar el análisis de los resultados de riesgo y el comportamiento de los mismos en los grupos a lo que se denomina post-procesamiento.

INTRODUCCIÓN

Aunque los SIG están creados para manipular datos espaciales, se demanda el uso de técnicas que permitan extraer conocimiento de estos datos acumulados en los Sistemas de Gestión de Bases de Datos Espaciales (SDBMS por sus siglas en inglés) y el descubrimiento de patrones que sean más fáciles de entender. Inicialmente se pudiera pensar en la extracción de conocimiento automatizado mediante la minería de datos, que permite encontrar conocimiento implícito en grandes volúmenes de datos (Pérez Betancourt y González Polanco 2013).

El descubrimiento de conocimiento o patrones en bases de datos espaciales a través de la minería de datos espaciales es complejo; pues no solo se encarga de los datos no espaciales, sino que además tiene en cuenta la localización de los objetos y sus relaciones topológicas. En este proceso se utilizan métodos basados en la generalización, en el reconocimiento de patrones, de agrupamiento: de exploración de asociaciones espaciales y mediante el uso de aproximación y agregación (Pérez Betancourt y González Polanco 2013; Oliva Santos, Macia Perez y Garea Llano 2014).

La minería de datos es un proceso de búsqueda de información relevante en grandes volúmenes de datos, semejante a la que podría realizar un experto humano (McDonnell, De la Fuente Aragón y McDonnell 2012; Kao et al. 2017). La amplia difusión de información espacial producto del desarrollo de los SIG ha favorecido la explotación de los datos con el objetivo de encontrar conocimiento de manera automatizada. La complejidad de los tipos de datos existentes en SDBMS y las estructuras de datos que las soportan limitan la utilización de técnicas tradicionales de minería de datos, lo que propicia la aparición de nuevas técnicas que de conjunto forman la minería de datos espaciales (Ester et al. 2000; Perumal et al. 2015; Kao et al. 2017). Esta se define además como el proceso automático o semiautomático de seleccionar, explorar, modificar, visualizar y valorar datos espaciales con el objetivo de descubrir conocimientos (Pérez Betancourt y González Polanco 2013).

El interés de la salud pública y la epidemiología en el estudio y análisis de la distribución geográfica de las enfermedades, su relación con los riesgos potenciales y el desarrollo de herramientas que permiten el manejo de datos epidemiológicos con su componente espacial, ha impulsado considerablemente el desarrollo de métodos estadísticos para ambas disciplinas. Como resultado de este planteamiento se consideran los avances en la elaboración de planes preventivos. Bajo estas circunstancias, la utilización de la minería de datos espaciales presenta muchas potencialidades para estudios epidemiológicos y de salud (García Pérez y Alfonso Aguilar 2013; Pérez Betancourt y González Polanco 2013; Pérez Betancourt, González Polanco, et al. 2016; Pérez Betancourt, González Polanco y Febles Rodríguez 2017).

INTRODUCCIÓN

Numerosas son las aplicaciones en las que se requiere del manejo de bases de datos espaciales con el objetivo de obtener conocimiento e identificar grupos. Estas son necesarias para descubrir importantes distribuciones del espacio en estudios salubristas, lo cual puede ser resuelto con el empleo de algún algoritmo de agrupamiento conveniente. Por lo tanto, el estudio, aplicación y creación de nuevos algoritmos constituye un desafío importante en la actualidad.

Como parte de los métodos utilizados en la minería de datos espaciales para el Descubrimiento de Conocimiento en Bases de Datos (KDD por sus siglas en inglés) se encuentran los de agrupamiento. El agrupamiento o clustering es una de las técnicas más útiles para encontrar conocimiento oculto en un conjunto de datos. En la actualidad el análisis de grupos en minería de datos se utiliza en una amplia variedad de áreas tales como: reconocimiento de patrones, análisis de datos espaciales, procesamiento de imágenes, cómputo y multimedia, análisis médico, economía, bioinformática y biometría principalmente (Han, Pei, & Kamber 2011; Agrawal et al. 2016; Gajewski y Martyn 2016).

El post-procesamiento es una parte importante y definitoria del proceso de estratificación de territorios ya que permite la obtención de conocimiento sobre un proceso determinado. Además es capaz de interpretar y evaluar el conocimiento extraído, visualizarlo o simplemente documentarlo para el usuario final (Yee Leung 2016). El post-procesamiento está encaminado a identificar los territorios y estratos más afectados así como determinar los posibles factores asociados al comportamiento de los mismos basándose en los factores de riesgo como parte de la cartografía de riesgos de cada territorio.

El concepto de cartografía de riesgos es un instrumento de enorme interés y aplicabilidad en la ordenación y planificación territorial, debido a que permite valorar el potencial riesgo del territorio para ubicar en él, diferentes actividades humanas: industria, servicios y fundamentalmente salud. La investigación sobre riesgos ha desarrollado conceptos de gran interés, como los de peligrosidad, exposición y vulnerabilidad, los denominados factores de riesgo, cuyas posibilidades operativas para un análisis territorial concreto ya han sido suficientemente exploradas en diversos estudios e investigaciones. Para que exista riesgo, necesariamente se tienen que dar todos los factores que se incluyen en su ecuación (Ayala 2002; Bosque 2013). De manera general, se puede decir que la cartografía de riesgos tiene como objetivo identificar las áreas geográficas susceptibles de sufrir daño en caso de que una amenaza se haga realidad (Ayala 2002; Bosque 2013).

Debido a la necesidad de analizar los datos espaciales mediante el agrupamiento de datos y extracción de conocimiento en la minería de datos espacio-temporales se introdujeron en Cuba los SIG y con ellos se han desarrollado herramientas que facilitan el proceso de estratificación territorial. El empleo de la

estratificación territorial en las áreas de la salud basado en algún indicador se presenta como el de más utilización en los estudios (López Caviedes 2004; García Pérez y Alfonso Aguilar 2013; Pérez Betancourt, Betancourt, et al. 2016).

En Cuba este proceso se desarrolla mediante soluciones informáticas por separado, primeramente se realiza el análisis estadístico y luego se presentan los resultados en mapas temáticos utilizando los SIG (Yenisei Bombino Companioni 2005). En el componente de estratificación en el que se basa la presente investigación (Morales Pérez y Vega Torres 2015) existen insuficiencias para realizar un post-procesamiento a los estratos y por tanto para obtener un diagnóstico sobre posibles factores de riesgo como parte de la cartografía de riesgos. Esto afecta cuando se requiere el análisis de la relación espacial de indicadores en diferentes áreas e influyen negativamente en la capacidad de gestión de las entidades.

Después de analizar la **situación problemática**, se identifica el siguiente **problema a resolver:**

Las insuficiencias en el componente de estratificación de territorios para realizar post-procesamiento de los estratos limitan la obtención de un diagnóstico sobre posibles factores de riesgo en los territorios.

Se delimitó como **objeto de estudio** el proceso de estratificación de territorios y se formuló la siguiente **idea a defender:**

Si se diseña una solución basada en un algoritmo de post-procesamiento de estratos que obtenga un diagnóstico se facilitará la información sobre posibles factores de riesgo en los territorios.

Para darle solución al anterior problema se plantea como **objetivo general** desarrollar un algoritmo de post-procesamiento de estratos para la obtención de un diagnóstico sobre factores de riesgo en los territorios. El **campo de acción** se delimitó en los algoritmos de post-procesamiento sobre estratos y el **objetivo general** se desglosó en los siguientes objetivos específicos.

Objetivos específicos:

1. Construir el marco teórico referencial de la investigación, relacionado con la minería de datos espaciales y el proceso de estratificación de territorios.

2. Diseñar un algoritmo de post-procesamiento para la obtención de un diagnóstico sobre posibles factores de riesgo en los territorios.

3. Implementar el algoritmo propuesto para el plugin de estratificación basado en indicadores de salud.

4. Validar la solución informática propuesta aplicando diferentes pruebas y métricas.

Estructura del trabajo

El presente trabajo está estructurado de la siguiente forma: introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas, glosario de términos y anexos. A continuación se muestra una breve descripción de cada uno de los capítulos.

Capítulo 1: Fundamentación teórica y metodológica de la minería de datos espaciales y la estratificación de territorios

En este capítulo se presentan elementos teóricos relacionados con los procesos de Minería de Datos Geoespaciales y la Estratificación de Territorios para lograr un mejor entendimiento del trabajo a desarrollar. Se tratan temas relacionados como las técnicas para el post-procesamiento, KDD, algoritmos de agrupamiento y cartografía de riesgos. Además, se realiza un estudio de la metodología, herramientas, tecnologías y lenguajes a utilizar en el desarrollo de la solución.

Capítulo 2: Algoritmo de post-procesamiento de estratos para la obtención de un diagnóstico sobre factores de riesgo

En este capítulo se realiza una descripción de la solución propuesta, se especifican los requisitos funcionales y no funcionales que se tendrán en cuenta para la implementación de la misma, y se detallan aspectos relacionados con su diseño y arquitectura. Se especifican los patrones del diseño aplicados y los artefactos derivados de la metodología de desarrollo de software seleccionada.

Capítulo 3: Verificación de la solución propuesta

Este capítulo describe la etapa de implementación, se elaboran y documentan las pruebas realizadas a la solución propuesta para demostrar el correcto funcionamiento de la misma, y por último se analizan los resultados obtenidos tras la aplicación de la herramienta en un caso de estudio.

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA Y METODOLÓGICA DE LA MINERÍA DE DATOS ESPACIALES Y LA ESTRATIFICACIÓN DE TERRITORIOS

En este capítulo se presentan un conjunto de elementos teóricos y metodológicos que conforman el marco referencial relacionado con el objeto de estudio. Se realiza un estudio del panorama actual del software SIG. Se realiza un análisis crítico de las herramientas existentes que soportan los procesos de estratificación de territorios, agrupamiento, minería de datos y KDD así como de los algoritmos de post-procesamiento. Por último, se analizan las tecnologías, herramientas informáticas y metodologías a utilizar en el proceso de desarrollo del software.

1.1 Estratificación

La estratificación se define como un conjunto de analogías que dan lugar a subconjuntos de unidades agregadas, denominadas estratos. Un estrato, por tanto, es un conjunto de unidades que presentan uno o varios parámetros, que los hacen similares entre sí y a la vez se diferencia de unidades correspondientes a otros estratos. Es decir, que en cada estrato existe una igualdad interna con diferencias o desigualdades externas (Batista Moliner et al. 2001; López Caviedes 2004; Acosta et al. 2013; Delgado Acosta et al. 2015).

La estratificación forma parte del proceso integrado de diagnóstico-intervención-evaluación, que como parte del enfoque epidemiológico de riesgo, es una estrategia útil para obtener un diagnóstico objetivo de acuerdo con el cual planificar las actividades de prevención y control de las distintas enfermedades. Sirve de base para categorizar metodológicamente e integrar áreas geoecológicas y grupos poblacionales de acuerdo a factores de riesgo (Pérez Betancourt, González Polanco y Febles Rodríguez 2017).

En esencia, la estratificación es la separación de datos en categorías o clases que permite aislar la causa de un problema, identificando el grado de influencia de ciertos factores en el resultado de un proceso.

La estratificación se describe en las siguientes fases (Cruz 2011; García Pérez y Alfonso Aguilar 2013):

1. Determinación del problema a estudiar.

2. Identificación y medición de las variables.

3. Aplicación del procedimiento de definición de estratos.

4. Identificación de los territorios y estratos más afectados.

5. Determinación de los posibles factores asociados al comportamiento.

6. Selección de intervenciones y adecuación de los servicios para la ejecución de las mismas.

7. Identificación de los indicadores de evaluación.

8. Ejecución de las intervenciones.

9. Evaluación de todo el proceso.

10. Monitoreo y ajuste de acuerdo con los problemas detectados.

La estratificación se usa con el objetivo de comprender de manera detallada la estructura de un grupo de datos, lo cual, permitirá identificar las causas del problema y llevar a cabo las acciones correctivas convenientes. La estratificación también permite examinar las diferencias entre los valores promedios y la variación entre diferentes estratos, y tomar medidas contra la diferencia que pueda existir.

1.1.1 Estratificación territorial en temas de salud

La estratificación territorial es un proceso que permite separar espacialmente los elementos representativos de los territorios (Batista Moliner et al. 2001). Su principal utilidad es identificar regiones de un país determinado en las cuales las condiciones de vida desiguales estén relacionadas con diferentes problemas de salud. En Cuba, su principal utilidad es identificar áreas con mayores necesidades de salud, con la finalidad de ofrecer a cada territorio de manera justa los recursos que realmente necesita y efectuar acciones específicas ante cada situación (Alegret Rodríguez, Herrera y Grau Abalo 2008; Ariadna y María del Carmen 2015).

Un elemento clave para la aplicación de la estratificación territorial es la precisión al evaluar los límites político-administrativos que demarcan los territorios en sus diferentes niveles (localidad, municipio, provincia, país) y su relación con la distribución de los problemas de salud. En este sentido, es importante destacar que los fenómenos y condiciones que afectan la salud responden a los factores que los originan y no necesariamente se distribuyen según esos límites territoriales (Pérez Betancourt, González Polanco y Febles Rodríguez 2017).

En el análisis de las condiciones de vida se evidencia esta dificultad, sin embargo la posibilidad de identificar de inmediato cuáles son las áreas geográficas que tienen peores condiciones de vida, así como la posibilidad de analizar cómo se presentan en estas los diferentes problemas de salud, resulta de extrema importancia para el Sistema de Salud; de manera que se utiliza la estratificación como una metodología muy eficaz para poner en evidencia estas desigualdades (Morales Pérez y Vega Torres 2015; Pérez Betancourt, González Polanco y Febles Rodríguez 2017).

Se identificaron en la bibliografía (Yenisei Bombino Companioni 2005; López Caviedes 2004) las técnicas de agrupamiento como unas de las más utilizadas y con mejores resultados para realizar la clasificación de los datos en los procesos de estratificación de territorios. Aunque las técnicas de agrupamiento están creadas para formar grupos con el fin de analizarlos, ver sus características y de forma general ayudar en el proceso de estratificación, los trabajos que acometen esta tarea carecen de post-procesamiento ejemplo de ello son (Yenisei Bombino Companioni 2005; Morales Pérez y Vega Torres 2015; Pérez Betancourt 2018). También se identificaron los SIG como herramientas para realizar el proceso de estratificación, que serán analizadas posteriormente.

1.2 Descubrimiento de conocimiento en bases de datos (KDD)

La identificación de patrones comunes, asociaciones, reglas generales y nuevo conocimiento es una actividad investigativa de gran interés, a este proceso se le denomina también KDD (Gilbert y Nonell 2005; Yee Leung 2016). Este es una respuesta a los enormes volúmenes de datos que se encuentran recopilados y almacenados en bases de datos operacionales y científicas. Por tales características puede ser sumamente útil la utilización de KDD en la estratificación de territorios.

KDD es el proceso de nivel superior para obtener información a través de la extracción de datos y la destilación de esta información en conocimiento (ideas y creencias sobre el mundo) a través de la interpretación de información e integración con el conocimiento existente. KDD se basa en la creencia de que la información está oculta en bases de datos muy grandes en forma de patrones interesantes (Yee Leung 2016).

La minería de datos y la minería de datos espaciales son el núcleo matemático del proceso KDD. Ambas técnicas forman parte de este proceso, que comprende los algoritmos que exploran los datos, desarrollan modelos matemáticos y descubren patrones significativos (implícita o explícitamente), los cuáles son la esencia del conocimiento útil (Peña Suarez 2017). Se les denomina patrones a las

relaciones que existen entre los elementos de los datos analizados. Los patrones son de interés, si son confiables, novedosos y útiles respecto al conocimiento que generan y el acoplamiento con los objetivos del análisis.

1.2.1 Métodos utilizados en la minería de datos espaciales para el descubrimiento de conocimiento

La minería de datos espaciales se utiliza para extraer conocimiento. Sus métodos pueden ser utilizados para explorar, descubrir relaciones entre datos espaciales y no espaciales, reorganizar datos espaciales en bases de datos y determinar sus características generales de manera simple. Existen diferentes métodos de minería de datos espaciales como (Cangrejo Aljure y Agudelo 2011; Pérez Betancourt y González Polanco 2013; Betancourt 2014; Peña Suarez 2017):

- Basados en el reconocimiento de patrones: son utilizados en la clasificación de información que pueden ser imágenes de satélites, fotografías, textos o cualquier fuente de datos:
- De agrupamiento: permiten agrupar los objetos de una base de datos en grupos llamados conglomerados, conformados por elementos tan similares como sea posible.
- De exploración de asociaciones espaciales: permiten descubrir reglas de asociación espacial que relacionen a uno o más objetos espaciales.
- Mediante el uso de aproximación y agregación: permiten descubrir conocimiento a partir de las características representativas de los objetos.

1.3 Minería de Datos Espaciales

La minería de datos espaciales posee la base teórica y metodológica para la identificación de patrones sobre los datos y tiene como objetivo descubrir de forma automatizada patrones inesperados potencialmente útiles en SDBMS. La minería de datos espaciales es considerada una rama de la minería de datos con la característica de extraer conocimiento referente a la naturaleza espacial de los datos (Cangrejo Aljure y Agudelo 2011; Pérez Betancourt y González Polanco 2013).

La minería de datos espaciales es la técnica de encontrar a través de diferentes métodos y herramientas patrones interesantes y previamente desconocidos, pero potencialmente útiles en bases de datos espaciales. Este tipo de bases de datos no almacenan explícitamente patrones o reglas que determinan las relaciones espaciales entre los objetos y algunas características no espaciales (Peña Suarez 2017).

1.3.1 Proceso de Minería de Datos

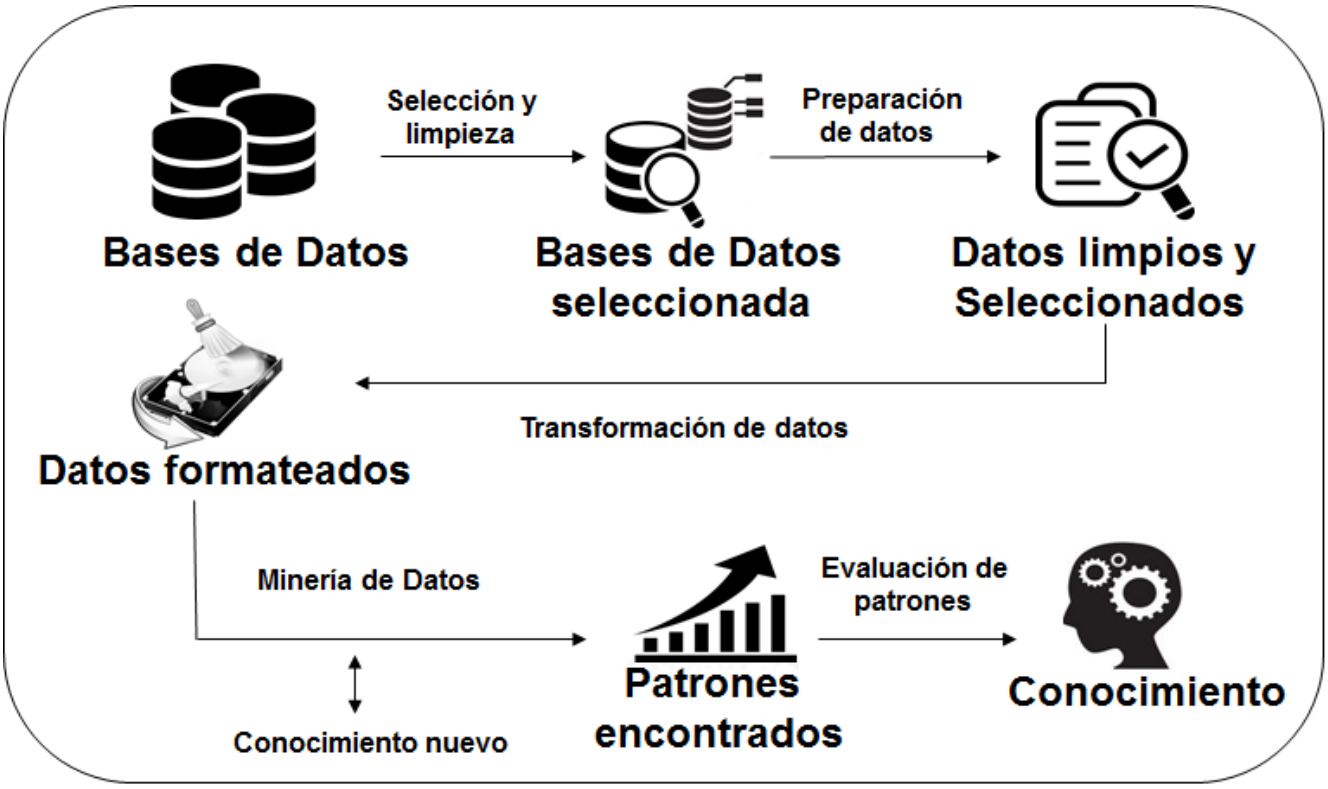

Figura 1 Proceso de Minería de Datos. Fuente: Elaboración Propia tomando como ejemplo (Peña Suarez 2017)

Un proceso típico de minería de datos consta de los siguientes pasos generales (Hernández, Ramírez , & Ferri 2007; Pérez Betancourt y González Polanco 2013):

- **Selección y limpieza**, tanto en lo que se refiere a las variables objetivo (aquellas que se quiere predecir, calcular o inferir), como a las variables independientes (las que sirven para hacer el cálculo o proceso), como posiblemente al muestreo de los registros disponibles.
- **Preparación de datos**, en especial los histogramas, diagramas de dispersión, presencia de valores atípicos y ausencia de datos (valores nulos).
- **Transformación de datos**, se realizará de diversas formas en función del análisis previo, con el objetivo de prepararlo para aplicar la técnica de minería de datos que mejor se adapte a los datos y al problema, a este paso también se le conoce como pre-procesamiento de los datos.
- **Minería de datos**, se construye el modelo descriptivo o predictivo, de clasificación o segmentación.
- **Conocimiento nuevo (extracción de conocimiento nuevo)**, mediante una técnica de minería de datos, se obtiene un modelo de conocimiento, que representa patrones de comportamiento

observados en los valores de las variables del problema o relaciones de asociación entre dichas variables. También pueden usarse varias técnicas a la vez para generar distintos modelos, aunque generalmente cada técnica obliga a un pre procesado diferente de los datos.

- **Evaluación de patrones**, una vez obtenido el modelo, se debe proceder a su validación comprobando que las conclusiones que arroja son válidas y suficientemente satisfactorias. En el caso de haber obtenido varios modelos mediante el uso de distintas técnicas, se deben comparar los modelos en busca de aquel que se ajuste mejor al problema. Si ninguno de los modelos alcanza los resultados esperados, debe alterarse alguno de los pasos anteriores para generar nuevos modelos.

1.3.2 Tareas de la Minería de Datos

Las tareas de minería de datos pueden ser de carácter descriptivo o predictivo. Las predicciones sirven para prever el comportamiento futuro de algún tipo de entidad mientras que una descripción puede ayudar a su comprensión. A continuación muestran las tareas de minería de datos (Cangrejo Aljure y Agudelo 2011):

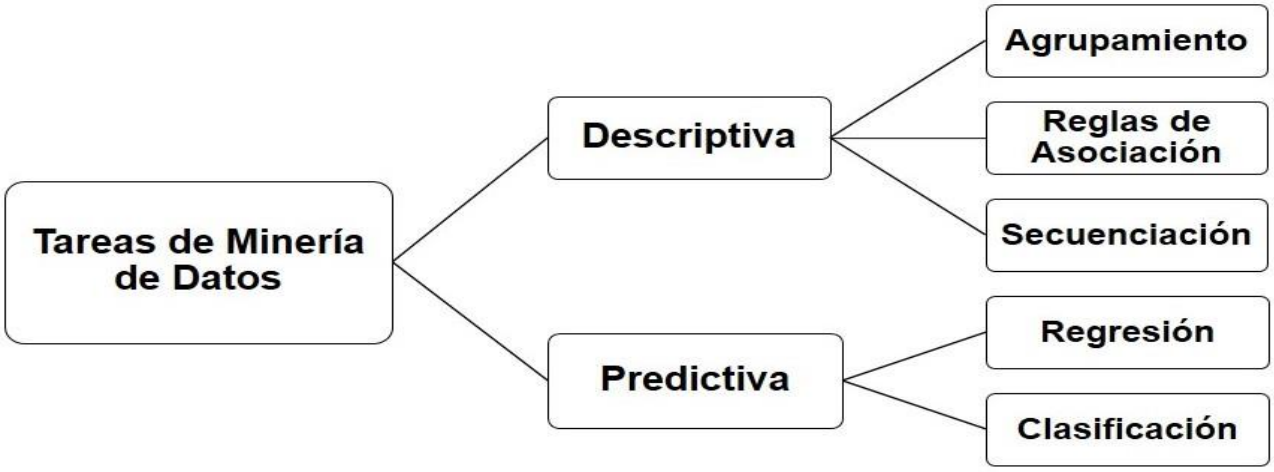

Figura 2 Tareas de la Minería de Datos.

Fuente: Elaboración Propia tomando como ejemplo (Peña Suarez 2017)

Predictiva

El objetivo de este tipo de minería, es predecir el valor particular de un atributo basado en otros atributos. El atributo a predecir es comúnmente llamado “clase” o variable dependiente, mientras que los atributos usados para hacer la predicción se llaman variables independientes.

Permite predecir valores de variables desconocidas (variable dependiente o variable objetivo) a partir de otros atributos de la base de datos (variables independientes) (Weiis y Indurkhya 1998).

- **Clasificación:** El objetivo de esta tarea es la clasificación de un dato dentro de las clases definidas del dominio que se está modelando. Permite la clasificación de los registros que tienen clase desconocida en categorías o clases ya definidas en la base de datos (Tan, Steinbach y Kumar 2006).
- **Regresión:** Predice un valor de una variable de valor continúo dado en base a los valores de las otras variables, suponiendo un modelo lineal o no lineal de dependencia. El objetivo es predecir los valores de una variable continua a partir de la evolución sobre otra variable continua, generalmente el tiempo. Ejemplo, se intenta predecir el número de clientes o pacientes, los ingresos, llamadas, ganancias, costos, etc. a partir de los resultados de semanas, meses o años anteriores (Hernández, Ramírez , & Ferri 2007).

Descriptiva

El objetivo de este tipo de minería, es encontrar patrones (correlaciones, tendencias, grupos, trayectorias y anomalías) que resuman relaciones en los datos. Se encarga de identificar patrones para la descripción de los datos existentes (Han, Pei, & Kamber 2011).

- **Agrupamiento:** Permite obtener grupos o conjuntos en donde se incorpore elementos similares extraídos de las clases del dominio dado (Riquelme, Ruiz y Gilbert 2006). Permite la segmentación en grupos excluyentes entre sí y cercanos dentro del grupo.
- **Reglas de Asociación:** Una asociación entre dos atributos ocurre cuando la frecuencia de que se den dos valores determinados de cada uno conjuntamente es relativamente alta. Ejemplo, en un supermercado se analiza si los pañales y la leche del bebe se compran conjuntamente (Hernández, Ramírez , & Ferri 2007). Encuentra relaciones entre dos o más atributos que ocurren con mayor frecuencia.
- **Secuenciación:** Es un conjunto de objetos dado, con cada objeto asociado con su propia línea de tiempo de eventos, encuentra reglas que predicen fuertes dependencias secuenciales entre los diferentes eventos. Las reglas se forman descubriendo primero patrones. Las ocurrencias de eventos en los patrones se rigen por restricciones de temporización (Hernández, Ramírez , & Ferri 2007).

A partir de las características de la minería de datos descriptiva (por agrupamiento) se propone el uso de la misma en la solución. Se tuvo en cuenta su habilidad para formar grupos y encontrar patrones significativos lo que facilita la obtención de un resultado más acertado.

1.3.3 Técnicas de Minería de Datos

Las técnicas de minería de datos permiten llevar a cabo las tareas predictivas y descriptivas haciendo uso de algoritmos de minería de datos. Según el objetivo del análisis de los datos, los algoritmos utilizados se clasifican en supervisados y no supervisados (Peña Suarez 2017)

- **Aprendizaje supervisado (o predictivo):** Predicen el valor de un atributo (etiqueta) de un conjunto de datos, desconocido a priori, a partir de otros atributos conocidos (atributos descriptivos). A partir de datos cuya etiqueta se conoce se induce una relación entre dicha etiqueta y otra serie de atributos. Esas relaciones sirven para realizar la predicción en datos cuya etiqueta es desconocida. Esta forma de trabajar se conoce como aprendizaje supervisado y se desarrolla en dos fases: Entrenamiento (construcción de un modelo usando un subconjunto de datos con etiqueta conocida) y prueba (prueba del modelo sobre el resto de los datos).
- **Aprendizaje no supervisado (o del descubrimiento del conocimiento):** Se descubren patrones y tendencias en los datos. El descubrimiento de esa información sirve para llevar a cabo acciones y obtener un beneficio (científico o de negocios) de ellas. Por estas características se decide la utilización para la investigación del aprendizaje no supervisado ya que contiene las técnicas de agrupamiento para la obtención de conocimiento**.**

Tabla 1. Técnicas de minería de datos (Peña Suarez 2017)

Supervisados	No supervisados
Arboles de decisión	Detección de desviaciones
Inducción neuronal	Segmentación
Regresión	Agrupamiento (clustering)
Series Temporales	Reglas de Asociación
	Patrones Secuenciales

1.3.4 Algoritmos de Minería de Datos Espaciales

Los algoritmos de minería de datos espaciales deben operar sobre conjuntos de datos de tamaño considerable, por lo que se debe trabajar en propuestas donde el conjunto de datos completo no resida en la memoria principal. Deben hacer un correcto uso de las técnicas de optimización de búsquedas espaciales y del razonamiento espacial y realizar su tarea de forma eficiente y rápida. A continuación, se describen algunos de los algoritmos más utilizados de la minería de datos espaciales (Pascual, Pla y Sánchez 2007; González 2010; José de Caldas 2016):

- **CLARANS:** consiste en la búsqueda aleatoria de un grupo de datos. Tiene complejidad temporal de O(n2). Producto de la importancia de los datos espaciales, este algoritmo se deriva del SD CLARANS, que busca descubrir características no espaciales en grupos espaciales, y del NSD CLARANS para descubrir conglomerados espaciales en grupos de datos no espaciales.
- **DBSCAN:** este algoritmo pertenece a la familia de algoritmos de conglomeración espacial. Aborda la integración entre la minería de datos espaciales y la interfaz con el sistema de bases de datos espaciales. No todos los datos deben permanecer en memoria principal y tienen un orden de ejecución de O(logn). Este algoritmo se basa en los conceptos de conglomerado, alcance directo por densidad, alcance por densidad y conexión por densidad.
- **ST-DBSCAN:** es un algoritmo de agrupamiento por densidad y basa su funcionamiento en el DBSCAN. Tiene la característica de descubrir grupos de acuerdo con valores no espaciales y espacio-temporales de los objetos. Tiene complejidad temporal O(n3).
- **PDBSCAN:** es un algoritmo de conglomeración paralelizable, que se basa en DBSCAN; utiliza una estructura de datos distribuida, basada en árboles de tipo R(dR*-tree) y curvas de Hilbert para encontrar puntos pertenecientes a los diferentes conglomerados en el momento de la partición del problema, pues permite que puntos espaciales cercanos se encuentren en la misma partición siempre que sea posible. Logra disminuir los tiempos de ejecución de algoritmos, como CLARANS, que tienen orden de ejecución cuadrático. Permite que el problema de la búsqueda de conglomerados en un conjunto de datos de gran tamaño sea paralerizable y tenga un tiempo de ejecución de O(logn).
- **K-Means**: es uno de los algoritmos particionales que sigue una forma sencilla para dividir una base de datos dada en *k* grupos (fijados a priori), donde cada grupo tiene asociado un centroide (centro geométrico del grupo). Los datos se asignan al grupo cuyo centroide esté más cerca (utilizando

cualquier métrica de distancia) y luego iterativamente, se van actualizando los centroides en función de las asignaciones de datos a grupos, hasta que los centroides dejen de cambiar.

Luego del análisis de los algoritmos de minería de datos espaciales se ha reportado en la literatura el uso de K-Means como uno de los más utilizados en los estudios. Este análisis se debe a que es uno de los algoritmos de agrupamiento más simples, conocidos y eficientes (Pascual, Pla y Sánchez 2007).

1.3.5 Bases de Datos Espaciales

Las bases datos espaciales, son un sistema administrador de bases de datos que maneja datos existentes en un espacio o datos espaciales. Estas bases de datos incluyen datos geográficos, imágenes médicas, redes de transporte o información de tráfico, etc., donde las relaciones espaciales son relevantes (Hernández, Ramírez , & Ferri 2007). En este tipo de bases de datos es imprescindible establecer un cuadro de referencia (un SRE, Sistema de Referencia Espacial) para definir la localización y relación entre objetos, ya que los datos tratados en este tipo de bases de datos tienen un valor relativo, no es un valor absoluto. Los sistemas de referencia espacial pueden ser de dos tipos (Pascual, Pla y Sánchez 2007; González 2010; José de Caldas 2016): georreferenciados (carreteras, ciudades, suelo, altitudes), son los que normalmente se utilizan, ya que es un dominio manipulable, perceptible y que sirve de referencia y no georreferenciados (son sistemas que tienen valor físico, pero que pueden ser útiles en determinadas situaciones), estos se almacena de dos formas: vectorial y ráster.

Una base de datos espacial puede reconocer y analizar las relaciones espaciales que existen en la información geográfica almacenada, mediante relaciones topológicas (Shekhar et al. 1999; Eldawy, Mokbel y others 2016). Estas relaciones permiten construir modelos y análisis espaciales complejos. En los sistemas de información geográfica se entiende como topología a las relaciones espaciales entre los diferentes elementos gráficos (topología de nodo/punto, topología de red/arco/línea, topología de polígono) y su posición en el mapa (proximidad, inclusión, conectividad y vecindad). Estas relaciones, que pueden ser obvias a simple vista, se deben establecer mediante un lenguaje y unas reglas de geometría matemática.

1.4 Técnicas de agrupamiento

En la literatura científica consultada (López Caviedes 2004; Yenisei Bombino Companioni 2005; Erik Limón 2012) se identificaron las técnicas de agrupamiento como unas de las más utilizadas y con mejores resultados para realizar la clasificación de los datos en los procesos de estratificación de

territorios. El proceso de agrupamiento consiste en la división de los datos en grupos de objetos similares. Las técnicas de agrupamiento se encargan de descubrir una estructura dentro de un conjunto de datos $D = \{x_i, \ldots, X_n\}$, dividiéndolo en subconjuntos que muestren ciertas coherencias. Es decir, los objetos pueden dividirse en grupos que contienen muestras similares dentro de un mismo grupo o clúster, más similares entre sí, que a las muestras de otros grupos; definiendo las muestras parecidas como una noción de similitud o de distancia entre muestras (Francisco José Cortijo Bon 2001; Jeon y Oh 2016; Abdullahi, Schardt y Pretzsch 2017).

Para medir la similitud entre objetos se suelen utilizar diferentes formas de distancia: distancia euclídea, de Manhattan, de Mahalanobis, entre otras (Wang et al. 2016). El representar los datos por una serie de clúster, conlleva la pérdida de detalles, pero consigue la simplificación de los mismos. Desde un punto de vista práctico, el agrupamiento juega un papel importante en aplicaciones de minería de datos, tales como exploración de datos científicos, recuperación de la información y minería de texto, aplicaciones sobre bases de datos espaciales (tales como GIS o datos procedentes de astronomía), aplicaciones web, marketing, diagnóstico médico, análisis de ADN en biología computacional, y muchas otras (Francisco José Cortijo Bon 2001; José de Caldas 2016).

La principal característica de estas técnicas es la utilización de una medida de similitud que, en general, está basada en los atributos que describen a los objetos y se define usualmente por proximidad en un espacio multidimensional. Para datos numéricos, suele ser preciso preparar los datos, de manera que se someten a un proceso de estandarización.

La actividad de agrupamiento implica los siguientes pasos (Jain, Murty y Flynn 1999; Hernández 2006).

- **Representación de patrones:** Se refiere al número de clases, número de patrones disponibles, y el número, tipo y tamaño de las características disponibles para el algoritmo de clustering.
- **Definición de proximidad:** La proximidad de los patrones es usualmente medida por una función distancia definida; esta función utiliza medidas de distancia como: euclidiana, manhattan, chebyshev y minkowski. (Gilbert y Nonell 2005).
- **Clustering o agrupamiento:** Puede ser realizado en un gran número de formas. Se pueden utilizar algoritmos de clustering jerárquicos, particionales y otras técnicas que abarcan métodos probabilísticos o de teoría de grafos.
- **Abstracción de datos:** Es el proceso de extraer una representación simple y compacta del conjunto de datos.

- **Verificación de resultados:** Consiste en validar el análisis de clustering realizado evaluando los resultados obtenidos.

Los métodos de agrupamiento no paramétricos pueden dividirse en tres grupos fundamentales: jerárquicos, particionales y basados en densidad. También existen otros como son los basados en grafos, basados en modelos, basados en restricciones y agrupamientos con alta dimensionalidad (Xu, Wunsch 2008), solo por mencionar los más empleados.

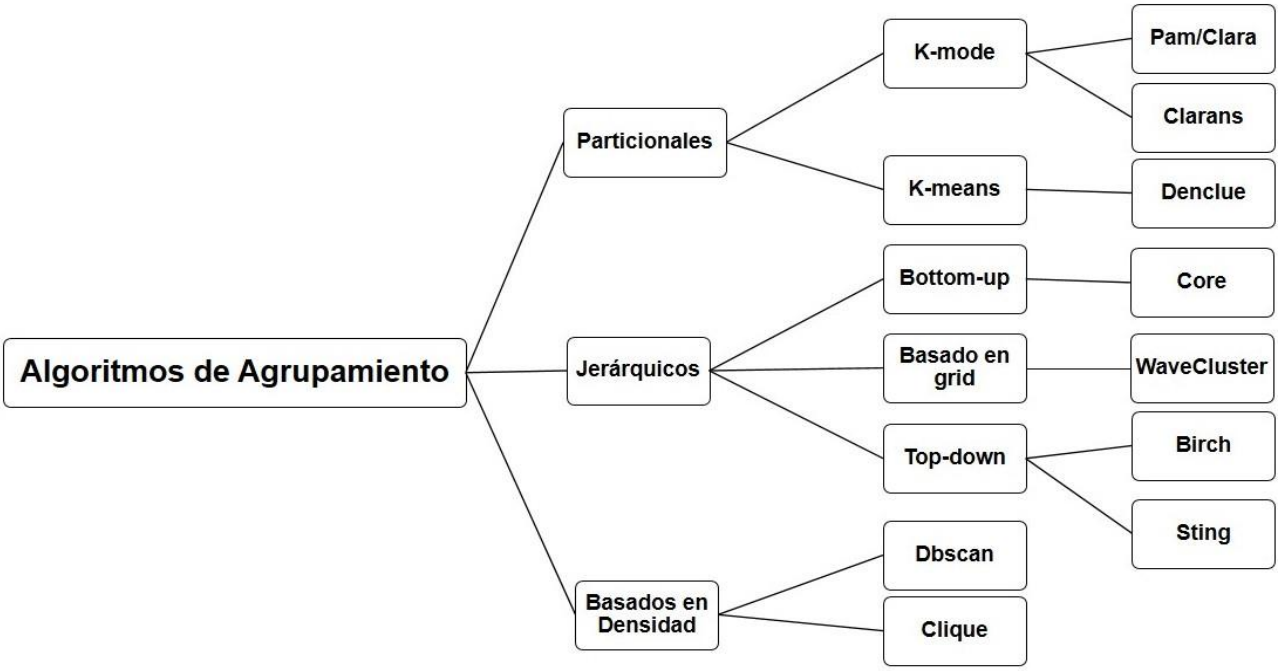

Figura 3. Algoritmos de agrupamiento.(Peña Suarez 2017).

1.4.1 Algoritmos jerárquicos

Los algoritmos jerárquicos son aquellos en los que se va particionando el conjunto de datos por niveles, de modo tal que en cada nivel generalmente , se unen o se dividen dos grupos del nivel anterior, según si es un algoritmo aglomerativo (de abajo hacia arriba) o divisivo (de arriba hacia abajo) (Feng, Wang y Chen 2014; Li, Li y Qiu 2017).

CORE es un algoritmo jerárquico muy eficiente y robusto (González 2010), el mismo utiliza una política mixta para el cálculo de la distancia entre dos grupos en cada iteración. Esta política es una especie de mezcla entre la política de centroides (donde la distancia entre dos clúster es la distancia entre sus centros de gravedad) y la llamada política del Minimum Spanning Tree (MST) (donde la distancia entre

dos grupos es igual a la distancia mínima entre dos puntos, uno en cada grupo) (Pascual, Pla y Sánchez 2007).

1.4.2 Algoritmos particionales

Los algoritmos particionales son los que realizan una división inicial de los datos en grupos y luego mueven los objetos de un grupo a otro según se optimice alguna función objetivo.

K-Means es un método de agrupamiento, que tiene como objetivo la partición de un conjunto de ***n*** observaciones en *k* grupos en el que cada observación pertenece al grupo cuyo valor medio es más cercano. Es un método utilizado en minería de datos (González 2010; José de Caldas 2016; Garre y Cuadrado 2017).

1.4.3 Algoritmos basados en densidad

Los algoritmos basados en densidad enfocan el problema de la división de una base de datos en grupos teniendo en cuenta la distribución de densidad de los puntos, de modo tal que los grupos que se forman tienen una alta densidad de puntos en su interior mientras que entre ellos aparecen zonas de baja densidad.

DBSCAN es uno de los primeros algoritmos de agrupamiento que emplea el enfoque de densidad (Pascual, Pla y Sánchez 2007). Comienza seleccionando un punto *t* arbitrario, si *t* es un punto central, se empieza a construir un grupo alrededor de él, tratando de descubrir componentes denso-conectadas; si no, se visita otro objeto del conjunto de datos (Francisco José Cortijo Bon 2001; Tsai y Chiang 2016; Hermawati y Sitanggang 2016).

1.4.4 Algoritmos basados en grafos

Los algoritmos de agrupamiento particionales y los basados en densidad comparten la desventaja de que el resultado final depende del orden en que se presentan los patrones al algoritmo de agrupamiento. Se puede argumentar en este sentido que el resultado será más acertado si todos los patrones pudieran considerarse simultáneamente. Los métodos de agrupamiento basados en grafos se basan en esta premisa, considerando las relaciones de similaridad entre todos los patrones empleados para el agrupamiento (Francisco José Cortijo Bon 2001; Pérez Betancourt y González Polanco 2013; Tian et al. 2014; Sun et al. 2016).

El algoritmo de agrupamiento basado en la matriz de similaridad es uno de los más sencillos consultados en la literatura (Francisco José Cortijo Bon 2001). Se basa en la construcción de una matriz de similaridad a partir de las distancias entre todas las parejas de patrones. La matriz de similaridad es una matriz cuadrada que se emplea para expresar el grado de similaridad entre cualquier pareja de patrones.

A partir de las características de las técnicas de agrupamientos analizadas se propone integrar el algoritmo basado en la matriz de similaridad a la propuesta de solución. Se tuvo en cuenta su sencillez y la característica de considerar todos los patrones simultáneamente para el agrupamiento, lo que permite obtener un resultado más acertado.

1.5 Sistemas de Información Geográfica como soporte para el proceso de estratificación

Los SIG poseen gran importancia tanto en la esfera social como económica, atendiendo además que la solución que se propone en esta investigación va encaminada a este tipo de sistemas, se hace imprescindible abordar los elementos fundamentales de los mismos.

Se puede definir un SIG como una integración de software, hardware y datos geográficos, diseñado para capturar, analizar, almacenar, manipular y desplegar información geográficamente referenciada (Chengfu, Xiaojun y Yujian 2010; Arsanjani et al. 2015; Jia et al. 2017). Puede definirse también como un modelo de una parte de la realidad referido a un sistema de coordenadas terrestres, construido principalmente para satisfacer la necesidad de información y ubicación geográfica del mundo. Los SIG son capaces de ubicar un objeto determinado en el espacio; encontrar donde está un cuerpo con respecto a otro; brindar información sobre su perímetro y área; encontrar el camino mínimo de un punto a otro, así como la generación de modelos a partir de fenómenos o actuaciones simuladas (Bravo 2000).

Actualmente, existe una gran diversidad de software SIG, cada uno de ellos con numerosas alternativas. Pudiendo resultar complejo elegir la adecuada a cada necesidad; para esto es necesario tener una visión global de sus representantes y de las características que los diferencian. Por este motivo fue realizado un breve análisis de algunas de las principales aplicaciones SIG libres por el grupo de investigación y desarrollo que creó la herramienta, a la cual, se le integra la presente solución. Para ello se consideró la característica más destacable del software libre para SIG: su modularidad (Víctor 2011); lo que favorece las interrelaciones y la reutilización de funcionalidades entre proyectos. Además, en el análisis se tuvo en cuenta el proceso de migración hacia software libre en el que se encuentran

inmersas las empresas cubanas. El resultado final del análisis arrojó que era factible la utilización del SIG QGis.

Software QGIS

Quantum GIS (QGIS, por sus siglas en inglés) es una aplicación SIG creada para la edición de mapas, multiplataforma y desarrollado utilizando Qt Toolkit [1] y C++. Ofrece muchas características SIG, entre las que se encuentran (QGIS DEVELOPMENT TEAM 2012; Pérez Betancourt 2016):

- Permite crear, editar, administrar y exportar mapas vectoriales en varios formatos.
- Permite realizar análisis de datos espaciales de PostgreSQL/PostGIS [2] usando el complemento de Python fTools[3].
- Incorpora a través de las herramientas de procesado, decenas de comandos de GRASS y SAGA para realizar análisis espacial tanto con datos vectoriales como ráster[4].
- Permite la integración de plugins [5] desarrollados en Python a través del módulo PyQGis.
- Presenta una interfaz amigable.

A partir de que la herramienta QGIS presenta diversas funcionalidades que pueden ser utilizadas en el desarrollo de la solución. Se tuvo en cuenta principalmente su capacidad en cuanto al manejo de la cartografía, lo que facilita la representación en mapas temáticos como resultado de la realización de procesos de estratificación territorial. Además se consideró que presenta una interfaz amigable, permitiendo agilizar el proceso de aprendizaje de la herramienta. Por último se identificó que cuenta con un módulo para la integración de complementos, permitiendo la reutilización de algunas de sus funcionalidades y la integración de la solución.

1.6 Herramientas, lenguajes y tecnologías a utilizar

En este como en todo proceso investigativo es necesario la utilización de sistemas de soporte que permitan organizar, facilitar, agilizar y automatizar las tareas generadas durante el transcurso de la

[1] Toolkit: Es una biblioteca multiplataforma para la creación de entornos gráficos.
[2] PostgreSQL/PostGIS: Ver Gestor de base de datos
[3] fTools: proporciona un recurso integral para muchas tareas comunes de SIG basados en vectores.
[4] Ráster: modelo de datos espaciales.
[5] Plugin: Es aquella aplicación que añade en un programa informático, una funcionalidad adicional o una nueva característica al software.

investigación. Las herramientas, lenguajes y tecnologías empleadas que se describen a continuación son de gran importancia para una correcta realización de la misma.

1.6.1 Lenguaje de modelado

UML es el acrónimo de Lenguaje Unificado de Modelado, este es el lenguaje estándar para visualizar, especificar, construir y documentar los artefactos de un sistema, utilizándose para el modelado del negocio y sistemas de software (Schefer-wenzl et al. 2013). También ofrece un estándar para describir los modelos, incluyendo aspectos conceptuales como procesos de negocio, funciones del sistema, expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

1.6.2 Herramienta CASE

CASE es el acrónimo de Computer Aided Software Engineering, las herramientas CASE son un conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software (Miguel Angel 2012).

Visual Paradigm

Es una herramienta de diseño UML y herramienta CASE UML diseñada para ayudar al desarrollo de software. Ofrece un paquete completo necesario para la captura de requisitos, la planificación del software, la planificación de pruebas, el modelado de clases y el modelado de datos (Started 2010).

Características principales de la herramienta (Started 2010):

- Soporta las últimas versiones del UML.
- Posee un generador de documentación y reportes en formato PDF, HTML y MS Word muy poderoso.
- Proporciona soporte para varios lenguajes en la generación de código e ingeniería inversa como: Java, C++, CORBA IDL, PHP, Ada y Python.
- Disponibilidad en múltiples plataformas (Windows, Linux)
- Capacidades de ingeniería directa e inversa.

Se selecciona Visual Paradigm for UML en su versión 8.0 como herramienta para el modelado UML, esta permite trabajar de forma colaborativa, hacer un trabajo organizado y ágil. Posibilita la realización de los diagramas necesarios para el desarrollo y mejor entendimiento de la aplicación. Permite realizar

ingeniería inversa a partir del código fuente. Al ser seleccionado el lenguaje de modelado UML, es conveniente tener en cuenta su vinculación con Visual Paradigm, resaltando que este último presenta abundantes tutoriales de UML y demostraciones interactivas.

1.6.3 Lenguaje de programación

Los lenguajes de programación son un conjunto de símbolos junto a un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones. Constan de un léxico, una sintaxis y una semántica (Louden 2004).

Partiendo de las características de la aplicación, se hace necesaria la selección de un lenguaje mediante el cual se pueda cumplir con los requisitos propuestos. Actualmente existen muchos lenguajes para el desarrollo de aplicaciones, surgidos a partir de las tendencias y necesidades de los escenarios. El análisis se centró fundamentalmente en el lenguaje Python.

Python

Se trata de un lenguaje interpretado o de script, con tipado dinámico, multiplataforma y orientado a objetos, que permite la programación imperativa, funcional y orientada a aspectos (Duque 2015).

Se seleccionó Python en su versión 2.7.6 porque su sintaxis es simple, clara y sencilla logrando de esta manera que los programas elaborados en este lenguaje parezcan pseudocódigo. Además el tipado dinámico, el gestor de memoria, la gran cantidad de bibliotecas disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo y rápido (Louden 2004).

Es importante tener en cuenta que al seleccionar QGIS como el software que soportará la integración de la solución, el lenguaje de programación más eficiente y conveniente para utilizar es Python; este SIG a partir de su versión 0.9 trae soporte del lenguaje Python que junto con el módulo PyQT4 entrega una solución óptima al desarrollo de plugins e interfaces gráficas de usuario. . Además de lo expuesto anteriormente, se escoge Python porque es el leguaje que se empleó para la solución en la cual se centra la presente investigación.

PyQt

PyQt es un conjunto de enlaces Python para la biblioteca gráfica Qt. El módulo está desarrollado por la firma británica Riverbank Computing y se encuentra disponible para Windows, GNU/Linux y Mac OS

bajo diferentes licencias. PyQt se distingue por su sencillez, por poseer un número importantes de herramientas que gestionen su manipulación y por su posibilidad de adecuarse a las distintas plataformas de software (Morales Pérez y Vega Torres 2015).

Con el empleo de PyQt en su versión 4.0 en el desarrollo de la herramienta informática, se podrá crear una interfaz visual sencilla ya que módulo posee los componentes visuales necesarios para su desarrollo y una abundante documentación.

Qt Designer

Qt Designer es una herramienta que permite acelerar el desarrollo de interfaces multilenguaje debido a que genera un archivo XML cuyo contenido es el formato de dicha interfaz, pudiéndolo convertir con los programas pertinentes a cada lenguaje. Esta herramienta provee características muy poderosas como la previa visualización de la interfaz, soporte para widgets y un editor de propiedades con gran variedad de opciones.

En correspondencia con la elección anterior de PyQt, se ha decidido emplear Qt Designer en su versión 4.7.4 como elemento que soportará el diseño de las interfaces. Su utilización permite la creación de las interfaces visuales de la aplicación de forma sencilla, además de la fácil manipulación de las variables de configuración de cada una de ellas (Morales Pérez y Vega Torres 2015).

1.6.4 Entorno de desarrollo integrado

Un entorno de desarrollo integrado (IDE, por sus siglas en inglés) es una herramienta que permite a los desarrolladores de software escribir sus programas en uno o más lenguajes. Consiste básicamente en una plataforma en la que se integran un editor de código, un compilador[6], un depurador[7] y una interfaz gráfica de usuario (Entornos de programación 2012).

Pycharm

Pycharm es un editor de código inteligente que proporciona soporte de primera clase para los lenguajes de programación: Python, JavaScript, CoffeeScript, TypeScript, HTML/CSS, Cython, lenguajes de plantilla, AngularJS y Node.js, y otros menos utilizados. Pycharm funciona en las plataformas Windows,

[6] Compilador: programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación.
[7] Depurador: programa usado para probar y eliminar los errores de otros programas.

Mac OS y Linux con una única clave de licencia, también ofrece un espacio de trabajo con colores personalizables y atajos de teclado (Duque 2015).

La decisión de seleccionar como IDE, Pycharm en su versión 3.4, está dada a que ofrece autocompletación inteligente de código, comprobación de errores sobre la marcha, soluciones rápidas y fácil navegación en el proyecto. Además, Pycharm mantiene la calidad del código bajo control con chequeos, asistencia a pruebas, refactorizaciones inteligentes, y una serie de inspecciones, lo que ayuda a escribir un código limpio y fácil de mantener (Jetbrains inc 2014).

1.6.5 Gestor de base de datos

Los Gestores de Bases de Datos (GBD) permiten crear y mantener una base de datos, además actúan como interfaz entre los programas de aplicación y el sistema operativo. El objetivo principal es proporcionar un entorno eficiente a la hora de almacenar y recuperar la información de las base de datos. Estos softwares facilitan el proceso de definir, construir y manipular bases de datos para diversas aplicaciones (Cobo 2007).

PostgreSQL

PostgreSQL es un sistema de GBD objeto-relacional, de propósito general, multiusuario y de código abierto, que soporta gran parte del estándar SQL[8]y ofrece modernas características como consultas complejas, disparadores, vistas, integridad transaccional, control de concurrencia multiversión. Puede ser extendido por el usuario añadiendo tipos de datos, operadores, funciones agregadas, funciones ventanas y funciones recursivas, métodos de indexado y lenguajes procedurales (PostgreSQL-3 Global Development Group 2014). Fue seleccionado PostgreSQL en su versión 9.0, teniendo en cuenta que es un GBD multiplataforma y de código abierto. Además se valoró la existencia de la extensión PostGIS para permitir el trabajo con datos espaciales.

PostGIS

Para añadir soporte a PostgreSQL de objetos geográficos se utilizó la herramienta PostGIS en su versión 2.1.5. Este módulo convierte la base de datos objeto-relacional PostgreSQL en una base de datos espacial para su utilización en SIG.

[8] SQL: lenguaje de consulta estructurado. Es un lenguaje declarativo de acceso a la base de datos.

PostGIS incluye un conjunto de operaciones para realizar consultas espaciales muy bien optimizadas por sus índices R-Tree [9] y su integración con el planificador de consultas de PostgreSQL. Utiliza las librerías Proj4 [10] para dar soporte a la transformación dinámica de coordenadas y la librería GEOS [11] para realizar operaciones de geometría. Utiliza bloqueo a nivel de fila, permitiendo a múltiples procesos trabajar con las tablas espaciales concurrentemente y asegurando la integridad de los datos (PostGIS development team 2014).

. **PgAdmin**

Como aplicación gráfica para gestionar el GBD PostgreSQL se utilizó la herramienta PgAdmin III en su versión 1.20.0. PgAdmin está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. Soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor y un agente para lanzar scripts programados. La conexión al servidor puede hacerse mediante conexión TCP/IP[12] o Unix Domain Sockets (en plataformas Unix), y puede encriptarse mediante SSL[13] para mayor seguridad (Robinson 2011).

1.7 Metodología de desarrollo

El desarrollo de un software no es una tarea fácil, se debe contar con un proceso bien detallado, para esto se necesita aplicar una metodología que sea capaz de llevar a cabo el control total del producto. Las metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un producto de software. Dichas metodologías pretenden guiar a los desarrolladores, sin embargo los requisitos de un software son muy variados y cambiantes, y se ha dado lugar a que exista una gran variedad de ellas (Letelier 2006).

9 R-Tree: es una estructura de datos de árboles usada para métodos de acceso espacial.

10 Proj4: biblioteca para realizar conversiones entre las proyecciones cartográficas.

11 Geos: librería para trabajar con datos geoespaciales.

12 TCP/IP: es una denominación que permite identificar al grupo de protocolos de red que respaldan a Internet y que hacen posible la transferencia de datos.

13 SSL: protocolo criptográfico que proporcionan comunicaciones seguras por una red.

Las metodologías se dividen en dos grupos, tradicionales (pesadas) y ágiles (ligeras). Las tradicionales, se centran en la definición detallada de los procesos y tareas a realizar, herramientas a utilizar, y requiere una extensa documentación, pretendiendo prever todo de antemano, además dependen de un equipo de desarrollo bastante grande. En las ágiles es más importante lograr que un producto de software se desarrolle con la calidad requerida, que realizar una buena documentación. En este tipo de metodología el cliente está presente en todo momento y colabora con el proyecto, que posee un equipo de desarrollo pequeño (Letelier 2006).

1.7.1 Programación extrema

Programación extrema (XP, por sus siglas en inglés) es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Además, se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. La metodología se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, donde existe un alto riesgo técnico (Joskowicz 2008).

Características de la metodología XP (Beck 2000):

- XP es una metodología “ágil” que no tiene en cuenta la utilización de elaborados casos de uso y la generación de una extensa documentación.
- XP tiene asociado un ciclo de vida y es considerado a su vez un proceso.
- La tendencia de entregar software en espacios de tiempo cada vez más pequeños con exigencias de costos reducidos y altos estándares de calidad.
- XP define Historias de Usuario (HU) como base del software a desarrollar, estas historias las escribe el cliente y describen escenarios sobre el funcionamiento del programa. A partir de las HU y de la arquitectura perseguida se crea un plan de liberaciones entre el equipo de desarrollo y el cliente.

Fases de la metodología XP:

- Planificación: Durante esta etapa se lleva a cabo el proceso de identificación y confección de las HU

- Diseño: Durante esta etapa se crea un diseño evolutivo que va mejorando incrementalmente y que permite hacer entregas pequeñas y frecuentes de valor para el cliente, basado principalmente en el desarrollo de las tarjetas Clase-Responsabilidad- Colaboración (CRC).
- Desarrollo: En esta fase se realiza la implementación de las HU que fueron seleccionadas por cada iteración. Al inicio se lleva a cabo un chequeo del plan de iteraciones por si es necesario realizar modificaciones. Como parte de este plan se crean tareas de ingeniería para ayudar a organizar la implementación exitosa de las HU.
- Pruebas: Esta fase permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones. XP divide las pruebas del sistema en dos grupos: pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores, y pruebas de aceptación o pruebas funcionales destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñada por el cliente final.

El ciclo de desarrollo consiste en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.

2. El programador estima el esfuerzo necesario para su implementación.

3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.

4. El programador construye ese valor de negocio.

5. Vuelve al paso 1

A partir del estudio de XP, se concluye que responde a las necesidades principales de tiempo, entorno y cantidad programadores, e incluye al cliente como parte fundamental del equipo de desarrollo. Además, se preocupa más en el avance exitoso del producto que en generar una documentación detallada del mismo, siendo capaz de adaptarse a los cambios de requisitos en cualquier punto del ciclo de vida del proyecto. Estos elementos demuestran que es una metodología factible para guiar el proceso de desarrollo de la solución, por lo que se decide incluir en la propuesta.

Conclusiones parciales

El desarrollo de este capítulo contribuyó a que se obtuviera un mayor enfoque y dimensionamiento del problema a través del análisis de los conceptos asociados a la solución como: KDD, minería de datos y agrupamiento. En función de los resultados obtenidos se llegó a las siguientes conclusiones:

- El estudio de las fases del proceso de estratificación proporcionó conocimiento preciso a la hora de tomar decisiones sobre métodos y vías de soluciones en cada una de dichas fases.
- En los estudios reportados en la literatura no se cubren totalmente las fases del proceso de estratificación de territorios.
- El estudio de algunos sistemas para la estratificación de territorios facilitó obtener una mejor visión para el post-procesamiento de estratos.
- Para la fase de implementación de la solución fue seleccionado un conjunto de herramientas y tecnologías basadas en licencias de software libre, con el objetivo de obtener un producto de alta independencia tecnológica y que sea utilizable en diferentes plataformas. Finalmente se escogió la metodología XP para guiar el proceso de desarrollo.

CAPÍTULO 2. ALGORITMO DE POST-PROCESAMIENTO DE ESTRATOS PARA LA OBTENCIÓN DE UN DIAGNÓSTICO SOBRE FACTORES DE RIESGO

En este capítulo se presenta una propuesta de solución para realizar el post-procesamiento en proceso de estratificación de territorios. Se especifican los requisitos de software y se obtienen los artefactos correspondientes a las fases de planificación y diseño de la metodología seleccionada. Además, se define la arquitectura y los principales patrones de diseño utilizados en el desarrollo de la solución.

2.1 Propuesta para la estratificación de territorios con post-procesamiento de estratos

En el presente trabajo el proceso de estratificación se desglosa en las fases siguientes, donde se incluye el pos-procesamiento de estratos como una nueva fase en la solución ya existente y que es a su vez el objetivo principal de la presente investigación.

Fases del proceso de Estratificación:

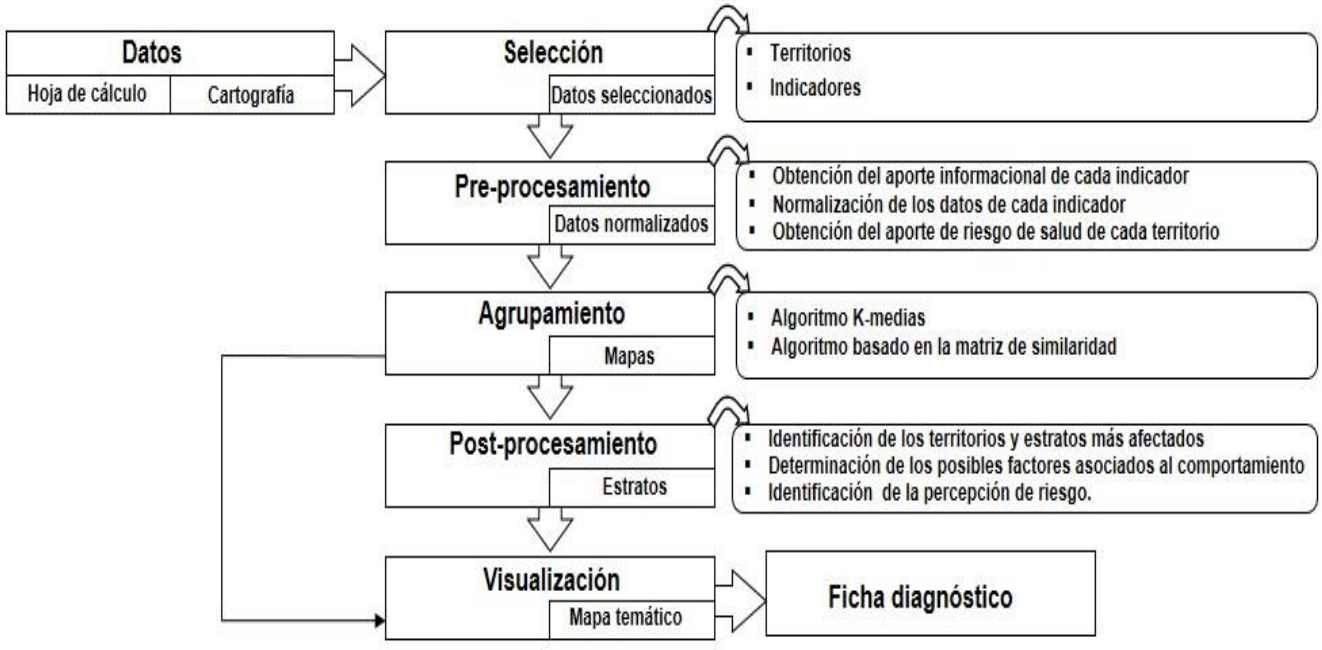

Figura 4. Modelo conceptual de la propuesta para la estratificación .Fuente: Elaboración propia.

En la figura 4 se presenta el modelo conceptual de la propuesta de solución. Para realizar el proceso de estratificación se propone la utilización de indicadores estadísticos seleccionados por el usuario y el empleo de la naturaleza espacial de los datos a través de indicadores cartográficos.

CAPÍTULO 2. ALGORITMO DE POST-PROCESAMIENTO DE ESTRATOS PARA LA OBTENCIÓN DE UN DIAGNÓSTICO SOBRE FACTORES DE RIESGO

Datos

Los datos son indicadores estadísticos tomados de orígenes de datos oficiales como el Anuario Estadístico emitido por la Oficina Nacional de Estadísticas e Información (INFOMED 2016) y la cartografía basada en una fuente cartográfica compuesta por capas, polígonos y puntos. Para realizar el proceso de estratificación se propone en (Pérez Betancourt, González Polanco, et al. 2016), la utilización de indicadores estadísticos seleccionados por el usuario y el empleo de la naturaleza espacial de los datos a través de los indicadores cartográficos siguientes:

- Cantidad de fuentes contaminantes
- Cantidad de ríos que presentan contaminación

Para la incorporación de estos dos indicadores se tuvo en consideración según (Pérez Betancourt, González Polanco, et al. 2016):

- El problema de contaminación de las aguas, se encuentra entre los principales problemas ambientales a los que se expone la sociedad, vinculado principalmente al impacto producido en áreas densamente pobladas y las alteraciones a la salud y a la calidad de vida de la población (Barceló and de Alda, 2008; Gil et al., 2012).
- El problema de la contaminación del aire, afecta a la sociedad y a la salud del ser humano. Según datos de la Oficina Nacional de Estadísticas se evidencia que cada año más del 30 por ciento de los cubanos sufren enfermedades respiratorias y otras asociadas con la contaminación del aire.
- La utilización de los de datos espaciales puntos y polilíneas permite servir de nivel de partida para incorporar el desarrollo de futuros trabajos, debido a que las distintos indicadores estudiados en la literatura están asociados a este tipo de datos.

Selección

Se eligen los factores de estratificación (territorios, indicadores).

Pre-procesamiento de los datos

Se obtiene el aporte informacional y se normalizan los datos de los indicadores seleccionados.

- **Obtención de la aportación informacional de los indicadores**

  La aportación informacional de los indicadores se obtiene mediante un método estadístico (Coeficiente de variación) que calcula el coeficiente de variación de cada uno. Debido a los diferentes dominios en los que se presentan los datos de los indicadores, se utiliza el coeficiente de variación (López Caviedes 2004).

  Posteriormente se obtiene la desviación estándar (S), la cual se obtiene de la raíz cuadrada de la varianza de los datos. El coeficiente de variación obtiene la dispersión de los datos en función de su promedio. Luego se realiza el proceso de normalización de los datos seleccionados para evitar que un indicador no domine sobre otro.
- **Medida de similitud empleada**

  Para determinar la semejanza entre los territorios se utiliza las métricas como distancia euclidiana ponderada, de Manhattan o de Mahalanobis. Para el presente trabajo se escogió la métrica distancia euclidiana ponderada. Esta medida de similitud se identifica como una de las más utilizadas y sencillas. En esta métrica cuando los valores son numéricos, se obtienen resultados satisfactorios en la clasificación (Rodríguez, Blanco y Camacho 2013).
- **Obtención del riesgo de salud por territorio**

  Debido a que los agrupamientos generados no tendrían razón alguna para el experto si no sabe cuál de estos grupos presenta mayor riesgo de salud, se propone desarrollar una funcionalidad que etiquete los grupos obtenidos por riesgo, dividido en dos casos de evaluación de la aportación de riesgo para los indicadores, debido a que en algunos indicadores un valor alto no significa que tenga mayor riesgo y viceversa.

Agrupamiento de los datos

En esta fase se clasifican los territorios en grupos homogéneos (estratos), para ello se utilizan los algoritmos de agrupamiento de datos antes expuestos. Como resultado se obtiene un mapa con los estratos geo-referenciados.

Post-procesamiento de estratos

En esta fase se determina los indicadores que afectan y los factores asociados al comportamiento de los mismos. Luego mediante el proceso de Percepción Geo-social se evalúan cada indicador que afecta un territorio especifico.

Visualización

Es la fase o etapa conclusiva del proceso de estratificación de territorios donde se representa en un mapa temático cada grupo homogéneo de territorios y se genera una Ficha Diagnóstico. Esta fase puede ejecutarse luego del agrupamiento o del post-procesamiento.

2.2 Post-procesamiento de estratos

Post-procesamiento en estratificación de territorios es la fase en la cual se reciben los datos resultantes del agrupamiento (mapa con los estratos) con el fin de obtener un diagnóstico final con los indicadores que más inciden sobre estos estratos. Esto quiere decir que una vez obtenida la información mediante el análisis que realiza el algoritmo se conoce cuáles son los estratos más afectados, así como, los indicadores que los afectan. Este dato permite que se haga un análisis de sentimiento en twitter, el cual, verifique el nivel de satisfacción de los usuarios con la veracidad del tweet sobre el indicador en cuestión, arrojando como resultado la ficha diagnóstico con la información precisa.

2.2.1 Identificación de los territorios y estratos más afectados

La identificación de los territorios y estratos más afectados se conoce luego del resultado del Pre-procesamiento (etapa de la estratificación que antecede al agrupamiento), luego de que en esta etapa se obtiene el riesgo de salud por territorio se puede saber el riesgo de salud también por estratos, los cuáles, están compuestos por dichos territorios. A continuación se presenta el método *obtenerTerritoriosxEstrato()* el cual retorna de cada estrato cuales territorios poseen un aporte de riesgo mayor que el promedio de riesgo del estrato.

Algoritmo *obtenerTerritoriosxEstrato()*

Entradas:

- Un conjunto de N patrones {N_1,…, Nm} # Territorios
- Un conjunto de K patrones {K_1,…, Kn} # Indicadores

Salidas:

- Los N= N_1, N_2 ,…,N_m que contenga T

Auxiliares:

- Lista T = {}
- Lista Ttemp= {}

Pasos:

1. M ← CalcularRiesgoProm()
2. Para cada N= N_1, N_2 ,…,N_m
3. N_i.calcularAporteRiesgo()
4. Ttemp[N_i] ← N_i.aporteRiesgo
5. Si N_i.aporteRiesgo > M
6. T[N_i] ← N_i.aporteRiesgo
7. Fin-Si
8. Fin-Para
9. Repetir el paso 2 hasta que la lista N sea recorrida por completo.
10. Si long[T] = 0
11. Para cada N= N_1, N_2 ,…,N_m en ordenarDeMayoraMenor(Ttemp)
12. T[N_i] = M
13. Si long[T] >= 4
14. Terminar
15. Fin-Si
16. Fin-Para
17. Repetir el paso 11 hasta la lista Ttemp sea recorrida por completo.
18. Fin-Si
19. Retornar T

Existe otro algoritmo que también aporta a esta etapa dentro de la solución desarrollada como es *TerritConTodosXEInd()* el cual retorna los territorios donde todos sus indicadores tengan un aporte de riesgo mayor que el promedio de riesgo de ese indicador.

2.2.2 Determinación de los posibles factores asociados al comportamiento de los estratos

En esta etapa en cuestión es donde se realiza todo el análisis estadístico que determina, en la relación indicador-estrato, cuál de los datos altera el valor de riesgo de salud de cada estrato. Para ello se necesita conocer de cada territorio de un estrato cual es el indicador que más incide en el valor de riesgo de salud, luego, de todos los indicadores que inciden en estos territorios saber cuál es el que más se repite o en consideración cuál es el que más afecta el valor de riego de salud del estrato.

CAPÍTULO 2. ALGORITMO DE POST-PROCESAMIENTO DE ESTRATOS PARA LA OBTENCIÓN DE UN DIAGNÓSTICO SOBRE FACTORES DE RIESGO

Para obtener los indicadores que más afectan al riesgo, se proponen utilizar los algoritmos de post-procesamiento de estratos que se describen a continuación:

1. El algoritmo *calcularMediasInd()* calcula el riesgo promedio de cada indicador y devuelve un diccionario de la forma (‘nombre de indicador’; promedio)

2. El algoritmo *indxEmediaEstrato()* basa su funcionamiento en encontrar de cada territorios los indicadores que tengan un aporte de riesgo mayor que el promedio de riesgo de ese indicador. Este algoritmo es capaz de reconocer los indicadores de ese territorio que están aportando al riesgo.

Algoritmo *calcularMediasInd()*

Entradas:

- Un conjunto de N patrones $\{N_1,\ldots, N_m\}$ # Territorios
- Un conjunto de K patrones $\{K_1,\ldots, K_n\}$ # Indicadores

Salidas:

- Los $K_1, K_2 ,\ldots,K_n$ que contenga M

Auxiliares:

- Lista M = {}
- Lista C = {}

Pasos:

1. Para cada N= $N_1, N_2 ,\ldots,N_m$
2. Para cada K= $K_1, K_2 ,\ldots,K_n$ en N.K
3. Si K_i está en la lista M
4. M[K_i.nombre] ← M[K_i.nombre] + M[K_i.aporteRiesgo]
5. C[K_i.nombre] ← C[K_i.nombre] + 1
6. Si-no
7. M[K_i.nombre] ← M[Ki.aporteRiesgo]
8. C[K_i.nombre] ← 1
9. Fin-Si
10. Fin-Para
11. Repetir el paso 2 hasta la lista K sea recorrida por completo.

12. Fin-Para
13. Repetir el paso 1 hasta la lista N sea recorrida por completo.
14. Para cada K= K_1, K_2 ,…,K_n en M
15. M[K_i] ← M[K_i] / C[K_i]
16. Fin-Para
17. Repetir el paso 14 hasta la lista M sea recorrida por completo.
18. Retornar M

Algoritmo *indxEmediaInd()*

Entradas:

- Un conjunto de N patrones {N_1,…, Nm} # Terrritorios
- Un conjunto de K patrones {K_1,…, Kn} # Indicadores

Salidas:

- Los K= K_1, K_2 ,…,Kn que contenga I

Auxiliares:

- Lista I = {}
- Lista IMedia= {}
- IndTemp= {}

Pasos:

1. IMedia ← calcularmediasDeLosIndicadores ()
2. Para cada N= N_1, N_2 ,…,N_m
3. Para cada K= K_1, K_2 ,…,Kn en N.K
4. Si K_i.aporteRiesgo >= IMedia [K_i.nombre]
5. Si N_i está en K
6. I [N_i].adicionar(K_i)
7. Si-no
8. IndTemp.adicionar(K_i)
9. I [N_i] = IndTemp
10. Fin-Si
11. Fin-Si

12. Fin-Para
13. Repetir el paso 3 hasta que la lista K sea recorrida por completo.
14. Fin-Para
15. Repetir el paso 2 hasta que la lista N sea recorrida por completo.
16. Retornar I

Existen otros algoritmos que también aportas a esta etapa dentro de la solución desarrollada como son: *IndDeTerritoriosXEMediaEstrato()* el cual retorna los territorios y los indicadores de esos territorios que tengan un aporte de riesgo mayor que el promedio de riesgo de del estrato y *IndMasAportanAlReisgoDelEstrato()* el cual obtiene de cada estrato los indicadores que tienen un promedio de aporte de riesgo mayor que el promedio de riesgo del estrato.

2.2.3 Percepción Geo-social

Twitter se ha convertido en una de las plataformas on-line más utilizadas para expresar opiniones e ideas; debido a esta razón que resulta una fuente de información para extraer estadísticas sociales. Por lo que se decide incorporar a la solución un método de análisis de sentimiento con el objetivo de encontrar contenido subjetivo en los textos de entrada y de extraer opiniones acerca de la información que aportan los algoritmos antes expuestos.

Twitter como método de obtención de datos

El estudio de redes sociales como herramientas de obtención de datos, ha supuesto un gran avance. Twitter ha abierto nuevas oportunidades de investigación y de negocio (Asensio Blasco 2014).

Uno de los temas más interesantes es el análisis de sentimiento y de opinión, donde se obtienen si los textos de los tweets contienen un sentimiento positivo, neutro o negativo (Asensio Blasco 2014). Existen mecanismos que además de palabras o expresiones de felicidad o de tristeza, buscan emoticonos del usuario y les asignan valor positivo o negativo.

Herramientas de análisis de datos geo-localizados (Asensio Blasco 2014)

- **Tweepsmap:** Ubica los seguidores de una cuenta en un mapa.
- **Trendsmap:** Geo-localiza las tendencias en tiempo real de cualquier lugar del mundo.
- **Twaps:** Busca los usuarios de Twitter y los 100 últimos tweets generados en un radio de dos millas alrededor de una localización indicada.

- **The One Million Tweet Map:** Geo-localiza en un mapa del mundo el último millón de tweets publicados, y se va actualizando en tiempo real. Otra de sus opciones interesantes es que también se puede filtrar por palabras clave o con hashtags, y ver dónde hablan de ese tema.

Qué es Twitter

Twitter es un servicio de microblogging que permite enviar mensajes de texto con un máximo de 140 caracteres. Estos mensajes se llaman tweets, y aparecen en la página principal del usuario. Cada usuario puede seguir a otros usuarios y ver sus tweets. (Asensio Blasco 2014)

Glosario de términos en Twitter (Asensio Blasco 2014)

- **Tweet (Tuit):** Publicación o actualización en Twitter.
- **Followers (Seguidores):** Usuarios que siguen una cuenta y leen los tweets que se envían.
- **Following (Seguidos):** Son las cuentas de Twitter que un usuario sigue. Los tweets publicados en las cuentas aparecen automáticamente en el timeline del usuario.
- **Timeline (TL):** Lista de tweets enviados de las cuentas que se siguen de manera cronológica.
- **Retweet (RT o retuit):** Función que permite volver a publicar un tweet, citando al usuario autor.

Qué información podemos extraer de Twitter (Asensio Blasco 2014)

Podemos dividir en cuatro secciones lo más relevantes de Twitter:

- **Quién**. La persona que lo escribió, o hizo un retweet, junto con sus datos públicos: nombre completo, localización, lenguaje, etc.
- **Cuándo**. Fecha y hora de publicación. Por ejemplo, con el perfil del usuario se puede determinar el horario en el que se encuentra.
- **Qué**. El contenido del tweet, que incluye el texto del mensaje además de los links, menciones o contenido multimedia.
- **Dónde**. Coordenadas geográficas de la ubicación desde donde fue publicado. Hay que tener en cuenta que no aparece en todos los tweets, sino que esta información es opcional.

Información extraíble de la API de Twitter (Asensio Blasco 2014)

Twitter proporciona su propio API (Application Programming Interface) oficial. Un API permite la comunicación entre diferentes componentes de software, añadiendo una capa entre ellos. El API de Twitter permite controlar tu cuenta y recuperar la información desde código (Asensio Blasco 2014).

Percepción Geo-social mediante el análisis de sentimientos en post-procesamiento

El análisis de sentimientos, también definido como minería de opción, es el procesamiento del lenguaje natural para identificar y extraer información subjetiva, información basada en el estado de ánimo de cada individuo. El análisis de sentimientos busca determinar la actitud del interlocutor con respecto a un tema específico o la polaridad contextual general de un documento. En específico busca conocer si lo que escribe el interlocutor es positivo o negativo, su impacto sobre el tema.

La determinación el concepto en sí del análisis de sentimientos realizado permite lograr determinar si los tweets extraídos sobre un tema específico contenían información positiva o negativa para su posterior clasificación y almacenamiento. Esto propicia la obtención de información directamente de los usuarios finales.

Este método en particular utiliza los indicadores que afectan los estratos y mediante los tweets relacionados realiza un análisis de sentimiento y de opinión de los usuarios que interactúan con los tweets así como sus datos y geo-localización, de esta forma se puede obtener información en porcientos de aceptación (positivo) o de inconformidad (negativo) de si verdaderamente este indicador está afectando el territorio específico.

2.2.4 Ficha diagnóstico

Luego de la implementación de los algoritmos propuestos, los datos arrojados por ellos son evaluados en el componente de percepción geo-social mediante el análisis de sentimiento utilizando twitter. Toda esta información debe ser registrada como parte del diagnóstico sobre posible factores de riesgo en los territorios. Es por ello que se construye la ficha diagnóstico que aportará información para la toma de decisiones, la cual está realizada a nivel de estrato y contiene los territorios que más aportan al riesgo, así como los indicadores que más influyen en el valor de riesgo de salud del estrato. Es importante decir que en esta ficha diagnóstico también se registra la percepción geo-social de los indicadores que más inciden en el valor de riesgo. A continuación se muestran los campos que componen dicha ficha:

- Ficha diagnóstico
- Número de estrato
- Riesgo de salud
- Territorios con mayor aporte al riesgo
- Indicadores con mayor aporte al riesgo

- Percepción geo-social positiva

La ficha diagnóstico servirá como guía a los especialistas para la toma de decisiones en materia de salud. Representa una herramienta de gestión crucial en el proceso de atención a los grupos poblacionales con mayor riesgo de enfermar o morir, ya que a través de ella se obtiene información sobre los aspectos socioeconómicos, demográficos y de salud que permitan, entre otros aspectos, la categorización de los territorios de acuerdo a los diferentes indicadores de salud.

2.3 Requisitos de software

"Un requisito es simplemente una declaración abstracta de alto nivel de un servicio que debe proporcionar el sistema o una restricción de éste" (Sommerville 2005). La calidad con que se realiza la captura de los requisitos afecta todo el proceso de desarrollo del software y repercute en el resto de las fases de desarrollo del mismo. Además, contribuye a tomar mejores decisiones de diseño y de arquitectura.

Existen los **requisitos funcionales** y los **no funcionales**, un requisito funcional (RF) define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Los requerimientos funcionales pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone que un sistema debe cumplir. Estos son complementados por los requisitos no funcionales, que se enfocan en cambio, en el diseño o la implementación (Sommerville 2005). Los requisitos no funcionales (RNF) son propiedades o cualidades que el sistema debe tener. Estas propiedades o cualidades se refieren a las características que hacen al sistema estable, usable, rápido, confiable y escalable (Sommerville 2005).

A continuación se muestran los RF identificados:

- **RF 1:** Obtener información de los territorios y estratos más afectados.
- **RF 2:** Construir reporte de los indicadores que afectan.
- **RF 3:** Importar fuente de datos.
- **RF 4:** Obtener percepción geo-social de los datos.
- **RF 5:** Construir ficha diagnóstico.
- **RF 6:** Visualizar resultados en mapa temático.
- **RF 7:** Visualizar ficha diagnóstico.

RNF identificados:

Requisitos de Software

- RNF 1: Se debe tener instalada la herramienta QGIS en su versión 2.6 o superior.
- RNF 2: Se debe tener instalado el GBD PostgreSQL en su versión 9.0 o superior.
- RNF 3: Se debe tener instalado el módulo Postgis en su versión 2.1.5 o superior.

Requisitos de Hardware

- RNF 4: La estación de trabajo debe contar con al menos 1,0 GB de Random Access Memory (RAM, por sus siglas en inglés).
- RNF 5: La capacidad mínima de espacio en disco debe ser 2.0 GB.

Requisitos de Usabilidad

- RNF 6: Debe tener una interfaz gráfica, visualmente atractiva para el usuario. La aplicación podrá ser usada por cualquier usuario con conocimientos básicos sobre geografía e informática. Debe mostrar mensajes al usuario que le ayuden a llevar a cabo la tarea que realiza.

Requisitos de Interfaz

- RNF 7: Debe tener una interfaz amigable y con apariencia profesional.
- RNF 8: La interfaz debe tener un diseño sencillo y ser de fácil comprensión para el usuario.

Restricciones de diseño e implementación

- RNF 9: Se hace uso de la herramienta QGIS en su versión 2.6 e IDE Pycharm 3.4.
- RNF 10: El lenguaje de programación usado para la implementación es Python.

2.4 Fase de planificación

La metodología XP define como fase inicial la planificación. Durante esta etapa se lleva a cabo el proceso de identificación y confección de las historias de usuario, así como la familiarización del equipo de trabajo con las tecnologías y herramientas seleccionadas para el desarrollo del software. El cliente

especifica la prioridad en que se deben implementar las historias de usuario, además de una estimación del esfuerzo. El resultado de la fase es un plan de entregas donde se realiza una estimación de las versiones que tendrá el producto en su realización, de manera tal que guíe su desarrollo.

Historias de usuarios

Las historias de usuarios (HU) es la técnica utilizada en XP para especificar los requisitos del software; en ellas el cliente describe brevemente las características que el sistema debe poseer, y se realiza una por cada característica principal del sistema. El tratamiento de las HU es muy dinámico y flexible, en cualquier momento pueden reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada HU es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en unas semanas (Letelier 2006).

Luego de obtener las principales funcionalidades del sistema, se identificaron 7 HU correspondientes a cada requisito funcional identificado. Se identificaron a partir de las HU seleccionadas 4 iteraciones. En las tablas 2 y 3 se muestra una breve descripción de dos de ellas, el resto se encuentran especificadas en anexos.

La estimación de los puntos de historia se desarrolló no solo para obtener un valor en horas de esfuerzo para el desarrollo de una historia de usuario (cada una de las parte en que se divide la funcionalidad a desarrollar en las metodologías ágiles) sino como una manera de dimensionar y relacionar la complejidad de las historias de usuario con respecto a otras. Para estimar los puntos de historias se escogió una historia de usuario de referencia a la que se le asignó un valor. Luego se analizan cada una y si se cree que representa el doble de trabajo que la historia de referencia, entonces el valor será de 2 puntos de historia; si se cree que tarda la mitad del esfuerzo, sería 0.5 puntos (Agile & Scrum 2016). En base a esta escala se propone que un punto de historia equivale a una semana de trabajo.

Tabla 2. Historia de usuario: Obtener información de los territorios y estratos más afectados.

Historia de Usuario "Obtener información de los territorios y estratos más afectados"	
Número: 1	**Nombre Historia de Usuario:** Obtener información de los territorios y estratos más afectados.
Usuario: Experto	
Prioridad en Negocio: Alto	**Riesgo en Desarrollo:** Alto
Puntos Estimados: 2	**Iteración Asignada:** 1

Programador responsable: Dalber I. Morán González
Descripción: La aplicación debe ser capaz de obtener información de los territorios y estratos más afectados así como cuáles son los indicadores que más afectan.
Observaciones: Se debe implementar un método que obtenga los territorios que más aportan al riesgo de cada estrato y retorne un listado de estos territorios. Se debe implementar un método que obtenga los territorios con todos sus indicadores con un promedio mayor que el aporte al riesgo de salud de ese indicador.

Tabla 3. Historia de usuario: Visualizar resultados en mapa temático.

Historia de Usuario "Visualizar resultados en mapa temático"	
Número: 6	**Nombre Historia de Usuario:** Visualizar resultados en mapa temático.
Usuario: Experto	
Prioridad en Negocio: Alto	**Riesgo en Desarrollo:** Alto
Puntos Estimados: 2	**Iteración Asignada:** 4
Programador responsable: Dalber I. Morán González	
Descripción: La aplicación debe ser capaz de mostrar una vista de la cartografía con los grupos creados y el valor de riesgo de los estratos generados.	
Observaciones: Se implementa una vista de la cartografía con los resultados del agrupamiento y el valor de riesgo de los estratos generados.	

Estimación de esfuerzos por historias de usuario

En el presente epígrafe se realiza la estimación del esfuerzo por HU, se hace necesario tener en cuenta que estas deben ser programadas en un tiempo de una a tres semanas. Si la estimación es superior a tres semanas, se divide en dos o más HU. Si es menor de una semana, se combina con otra HU. Estas estimaciones permiten tener una medida de la velocidad del proyecto y ofrecen una guía a la cual ajustarse. Los resultados estimados se muestran en la tabla 4.

Tabla 4. Estimación de esfuerzos por Historia de Usuario.

Historia de usuario	Puntos de estimación (semanas)

HU 1: Obtener información de los territorios y estratos más afectados.	2
HU 2: Construir reporte de los indicadores que afectan.	1
HU 3: Importar fuente de datos.	1
HU 4: Obtener Percepción geo-social de los datos.	2
HU 5: Construir ficha diagnóstico.	2
HU 6: Visualizar resultados en mapa temático.	2
HU 7: Visualizar ficha diagnóstico.	1

Plan de iteraciones

Una vez finalizadas las HU se debe crear un plan de iteraciones, indicando cuáles se desarrollarán en cada iteración. Para definir cuáles historias de usuario se desarrollarán en cada una de las iteraciones, se agruparon por las etapas del post-procesamiento y por la complejidad de las mismas. En la tabla 5 se muestra cómo quedó definido el plan de iteraciones de la solución propuesta.

Tabla 5. Plan de duración de las iteraciones.

Iteraciones	Orden de las historias de usuario a implementar	Duración de las iteraciones (semanas)	Criterio de culminación de la iteración
Iteración 1	**HU 1:** Obtener información de los territorios y estratos más afectados.	3	Al concluir esta iteración se obtienen los métodos que calculan los indicadores y territorios que aportan al riesgo de salud así como los estratos más afectados.
	HU 2: Construir reporte de los indicadores que afectan.		
Iteración 2	**HU 3:** Importar fuente de datos.	3	Al concluir esta iteración se obtiene los métodos que calculan la percepción social de los indicadores.
	HU 4: Obtener Percepción geo-social de los datos.		
Iteración 3	**HU 5:** Construir ficha diagnóstico.	2	Al concluir esta iteración se obtiene una integración de reportes con información de los indicadores que aportan al riesgo antes elaborados.

Iteración 4	**HU 6:** Visualizar resultados en mapa temático. **HU 7:** Visualizar ficha diagnóstico.	3	Al concluir esta iteración se obtiene una vista de un mapa con los estratos y una vista con información de los indicadores.
Total		11	

Plan de entrega

El plan de entregas establece qué HU serán agrupadas para conformar una entrega, y el orden de las mismas (Joskowicz 2008). En este plan se concentran las funcionalidades referentes a un mismo tema en módulos, esto permite un mayor entendimiento en la fase de implementación. Tiene como objetivo definir el número de liberaciones que se realizarán en el transcurso del proyecto y las iteraciones que se requieren para desarrollar cada una. De esta forma se puede trazar el plan de entrega en función de estos dos parámetros: el tiempo de desarrollo ideal y el grado de importancia para el cliente. En la tabla 6 se presenta el plan de entregas de la aplicación informática propuesta.

Tabla 6. Plan de duración de las entregas.

	Final de la 1ra Iteración	**Final de la 2da Iteración**	**Final de la 3ra Iteración**	**Final de la 4ta Iteración**
Módulos	1ra semana de marzo	4ta semana de marzo	2da semana de abril	1ra semana de mayo
Algoritmo de Post-procesamiento	v1.0	v1.1	v1.2	finalizado

2.5 Fase de diseño

La metodología de desarrollo XP plantea prácticas especializadas que accionan directamente en la realización del diseño para lograr un sistema robusto y reutilizable. Se trata en todo momento de conservar su simplicidad, es decir, crear un diseño evolutivo que va mejorando incrementalmente y que permite hacer entregas pequeñas y frecuentes de valor para el cliente, basado principalmente en el desarrollo de las tarjetas *Clase-Responsabilidad-Colaboración* (CRC).

Tarjetas Clase-Responsabilidad-Colaboración

Las tarjetas CRC son utilizadas para representar las responsabilidades de las clases y sus interacciones. Estas tarjetas permiten trabajar con una metodología basada en objetos, permitiendo que el equipo de desarrollo completo contribuya en la tarea del diseño. En cada tarjeta CRC el nombre de la clase se coloca a modo de título, las responsabilidades se colocan a la izquierda y las clases que se implican en cada responsabilidad a la derecha, en la misma línea que su requerimiento correspondiente.

Una clase es cualquier persona, evento, concepto, pantalla o reporte. Las responsabilidades de una clase son las cosas que conoce y las que realizan, sus atributos y métodos. Los colaboradores de una clase son las demás clases con las que trabaja en conjunto para llevar a cabo sus responsabilidades (Casas y Reinaga 2008).

En las tablas 7 y 8 se muestran las tarjetas CRC correspondientes a las clases *Estrato* y *Estratificación.* Se eligió representar las tarjetas CRC de estas clases debido a la importancia de las mismas en la solución. En la clase Estrato se implementaron los métodos del proceso de post-procesamiento.

Tabla 7. Tarjeta CRC para la clase Estrato.

Clase: Estrato	
Responsabilidad	Colaboración
• Calcular el aporte de riesgo de cada estrato. • Calcular los indicadores y territorios que más aportan al riesgo de cada estrato. • Crear instancias de la clase Territorio.	Estratificación, Territorio

Tabla 8. Tarjeta CRC para la clase Estratificación.

Clase: Estratificación	
Responsabilidad	Colaboración
• Crear instancias de la clase Estrato.	ControladorEstratificador, Estrato

2.5.1 Arquitectura

La arquitectura de software es la definición y estructuración de una solución que cumple con los requisitos técnicos y operativos. Optimiza atributos que implican una serie de decisiones, tales como la seguridad, el rendimiento y la capacidad de administración. Estas decisiones en última instancia, afectan la calidad de la aplicación, el mantenimiento, el rendimiento y el éxito global (Pressman 2005).

Estilo arquitectónico a utilizar

Un estilo es un concepto descriptivo que define una forma de articulación u organización arquitectónica. El conjunto de los estilos cataloga las formas básicas posibles de estructuras de software. Estos permiten expresar un esquema de organización estructural esencial para un sistema de software (Pressman 2005).

Arquitectura en capas

La arquitectura en capas se define como una organización jerárquica donde cada capa proporciona servicios a la inmediatamente superior y se sirve de las prestaciones que le brinda la inmediatamente inferior. Con esto se logra abstraer las funcionalidades de una capa de manera tal que pueda ser totalmente remplazada sin afectar a las otras, solamente cambiar las referencias de las implicadas en el cambio (Juan Peláez 2009). En la figura 5 se presenta una imagen de la arquitectura de la solución.

Estratificación

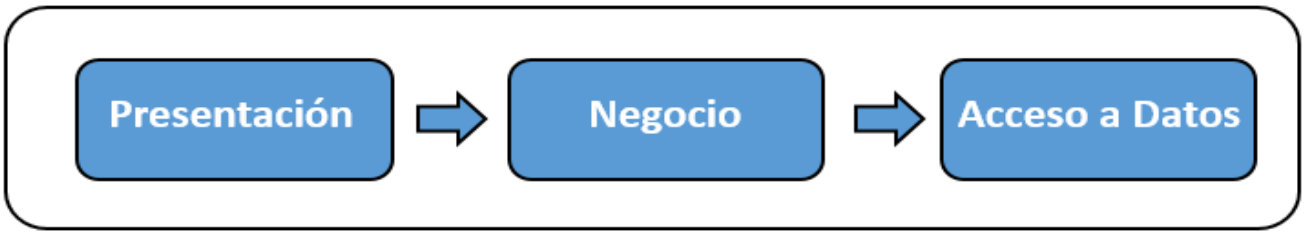

Figura 5. Evidencia de la arquitectura en capas.

Capa de presentación: es la parte de la aplicación con que el usuario interactúa, por lo que deberá cumplir muchos requisitos. Estos requisitos abarcan factores generales como la facilidad de uso, rendimiento, diseño e interactividad. Es importante que la aplicación tenga un buen diseño para apoyar una experiencia de usuario intuitiva, desde el principio, ya que la experiencia del usuario es influenciada por muchos aspectos diferentes de la arquitectura de la aplicación. En esta capa se encuentra la vista

del proceso de post-procesamiento con su respectiva UI_Postprocesamiento que conforman la vista de la solución.

Capa de negocio: es donde residen las clases gestoras de la información, se reciben las peticiones del usuario y se envían las respuestas a la capa de presentación. Se nombra capa de negocio porque es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes del usuario y presentar los resultados obtenidos, y con la capa de acceso a datos, para enviar datos que necesitan persistirse en la base de datos o recibirlos de la misma. En esta capa se encuentra la clase Estrato en la cual se implementaron los métodos de post-procesamiento.

Capa de acceso a datos: está constituida por las clases gestoras del acceso a datos, encargadas de acceder a los mismos y realizan todo el almacenamiento de la información. Esta capa recibe solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

2.5.2 Patrones de diseño

Los patrones de diseño constituyen la base para la búsqueda de soluciones a problemas comunes en el desarrollo de software y otros ámbitos referentes al diseño de interacción o interfaces. Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea considerada un patrón debe poseer ciertas características. Una de ellas es que debe haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores. Otra es que debe ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias (Craig 1999).

Patrones Generales de Software para la Asignación de Responsabilidades

Los Patrones Generales de Software para la Asignación de Responsabilidades (GRASP, por sus siglas en inglés) son utilizados para describir los principios fundamentales del diseño y la asignación de responsabilidades (Craig 1999). Entre los que se utilizaron en la solución figuran los siguientes: Experto, Creador, Controlador, Bajo acoplamiento y Alta cohesión.

Experto: El patrón Experto define como asignar de forma adecuada las responsabilidades en un modelo de clases. Indica que la responsabilidad de la creación de un objeto o la implementación de un método debe recaer en la clase que conoce toda la información necesaria para crearlo. Dicho patrón se evidencia en la aplicación informática, en la clase Territorio, como esta posee toda la información

necesaria para calcular el aporte de riesgo de cada territorio se le es asignada dicha responsabilidad. En la figura 6 se muestra una imagen de dicha clase.

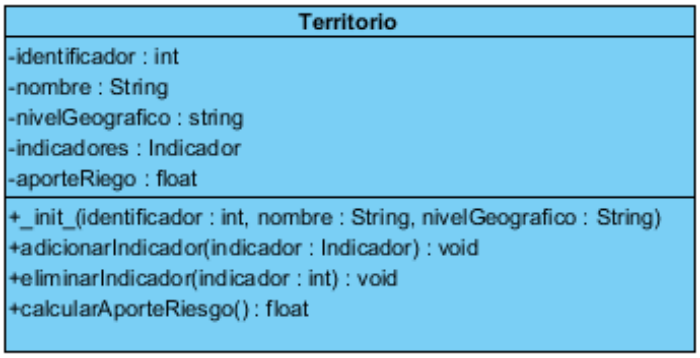

Figura 6. Evidencia del patrón Experto.

Creador: Este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos. La intención básica del mismo es encontrar un creador que necesite conectarse al objeto creado en alguna situación. En la aplicación informática se pone de manifiesto dicho patrón en la clase Estrato, a esta se le asigna la responsabilidad de crear instancias de la clase Territorio. En la figura 7 se muestra un ejemplo de dicho patrón.

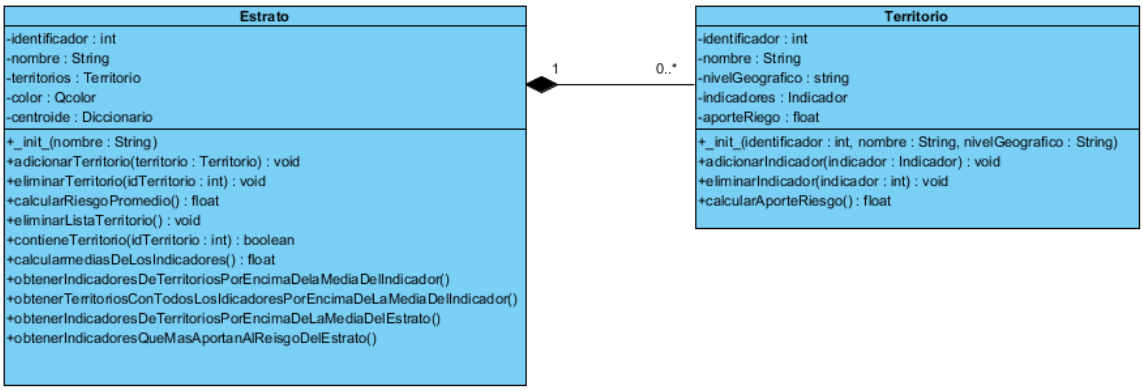

Figura 7. Evidencia del patrón Creador.

Bajo acoplamiento: Este patrón se garantiza en la aplicación basándose en la propia arquitectura del sistema, lo que permite que las dependencias entre las clases sea muy poca, ya que solamente las clases de una capa se pueden comunicar con las de la capa inmediatamente inferior. Este patrón se evidencia en la aplicación informática mediante la clase Estrato que contiene los métodos que componen el post-procesamiento y la vista llamada VistaPostProcesamiento.

Alta cohesión: La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Una baja cohesión hace muchas cosas no afines o realiza trabajo excesivo. En resumen, este patrón se observa cuando una clase tiene la responsabilidad de realizar una labor dentro del sistema, no desempeñada por el resto de los componentes del diseño. Este patrón se evidencia en la aplicación informática de forma tal que cada clase realice sus acciones, como es el ejemplo de la clase Estrato que evita que otra clase realice las acciones que deben ser acometidas por ella.

Controlador: Permite asignar la responsabilidad de controlar el flujo de *eventos del sistema*[14], a clases específicas, facilitando la centralización de actividades. El controlador no realiza estas actividades, las delega en otras clases con las que mantiene un modelo de alta cohesión. Un error muy común es asignarle demasiada responsabilidad y alto nivel de acoplamiento con el resto de los componentes del sistema. Este patrón se evidencia en la aplicación informática en la clase *ControladorEstratificador,* a esta se le asignó la responsabilidad de manejar los eventos del sistema generados por el usuario. En la figura 8 se muestra una imagen de dicha clase.

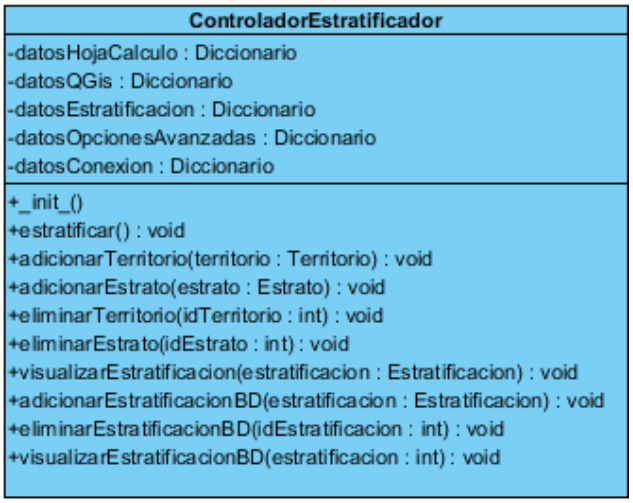

Figura 8. Evidencia del patrón Controlador.

[14] Evento del sistema: Es un evento de alto nivel generado por un actor externo. Es un evento de entrada externa.

Patrones del Grupo de Cuatro

Los Patrones del Grupo de Cuatro (GoF, por sus siglas en inglés) resuelven problemas específicos de diseño de software (Eric Rodríguez 2010). Estos patrones se agrupan en las siguientes categorías: creacionales, estructurales y de comportamiento.

- **Comportamiento:** Contribuyen a definir la comunicación e interacción entre los objetos de un sistema (Eric Rodríguez 2010).
- **Creacionales:** Permiten abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados (Eric Rodríguez 2010).
- **Estructurales:** Describen cómo las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades (Eric Rodríguez 2010).

Método plantilla: es un patrón de comportamiento que define en una operación el esqueleto de un algoritmo, delegando en las subclases algunos de sus pasos, esto permite que las subclases redefinan ciertos pasos de un algoritmo sin cambiar estructura. Este patrón se evidencia en las clases *AlgoritmoKMedias* y *AlgoritmoMatrizSimilitud*, estas heredan todas las funcionalidades de la clase *Algoritmo,* y redefinen los métodos *distancia()* y *run()* en función de sus características. En la figura 9 se muestra cómo se evidencia el patrón plantilla en la aplicación informática propuesta.

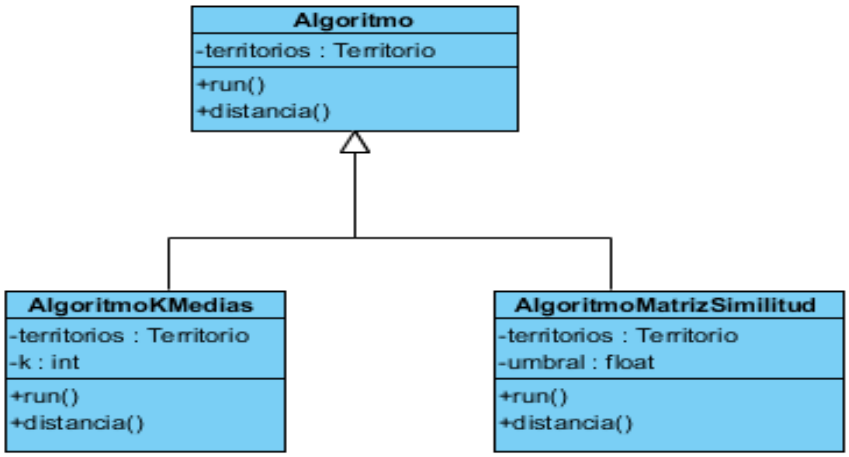

Figura 9. Evidencia del patrón Plantilla.

2.5.3 Modelo de la vista lógica de la estructura del sistema

El modelo organiza una descripción de la arquitectura de software utilizando la vista lógica. Ofrece soporte a los requerimientos funcionales, lo que el sistema debe proveer en términos de servicios a sus usuarios (Barrera León 2011). En la figura 10 se muestra el modelo de la vista lógica de la estructura del sistema de la aplicación informática propuesta.

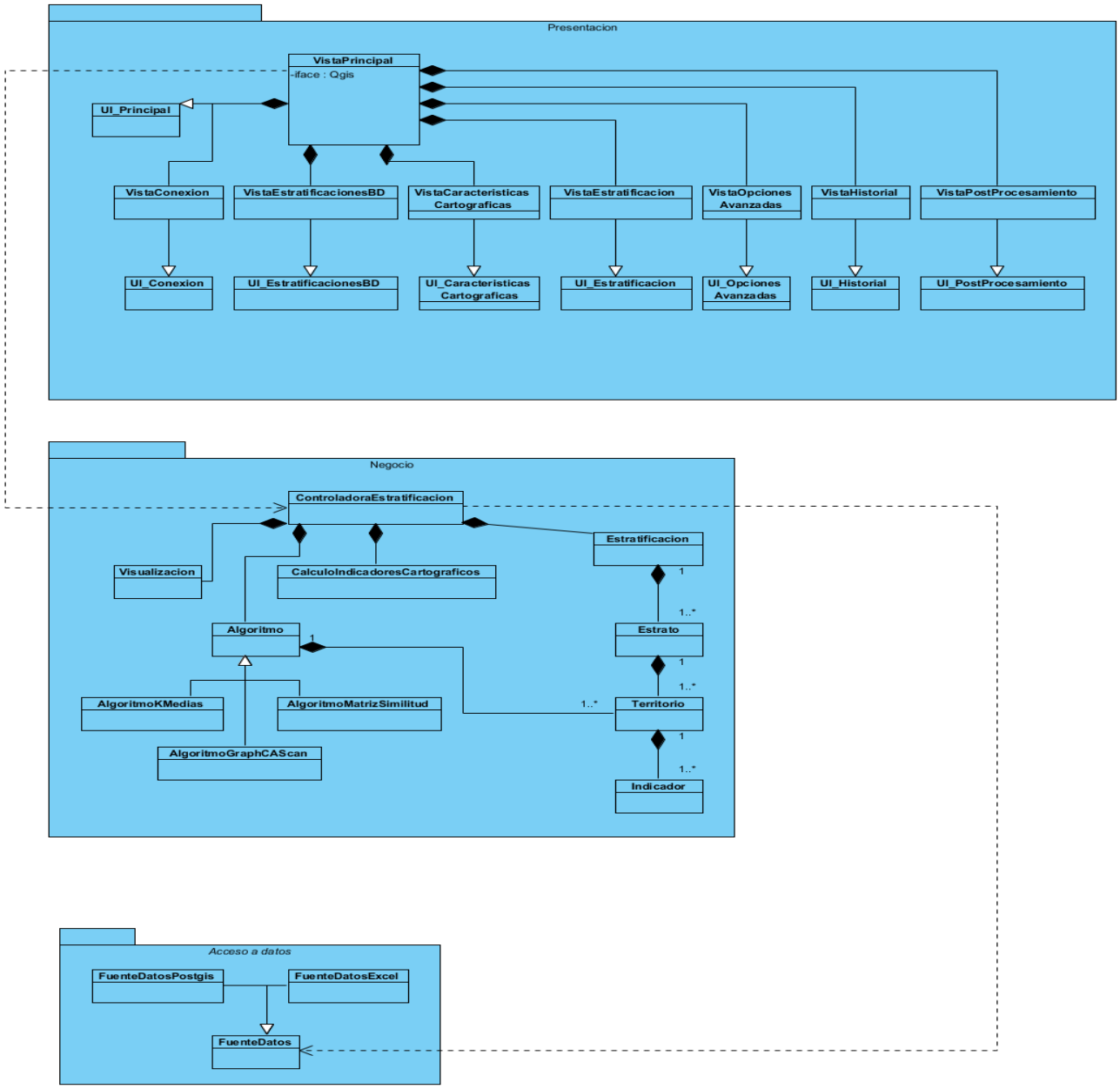

Figura 10. Modelo de la vista lógica de la estructura del sistema.

Conclusiones parciales

La propuesta de solución definida facilitó realizar el Post-procesamiento de estratos en el proceso de estratificación de territorios utilizando SIG. Luego de analizar los resultados obtenidos se llegó a las siguientes conclusiones:

- El algoritmo de post-procesamiento de estratos desarrollado permite dar solución a la insuficiencia de métodos que realicen esta tarea en los trabajos consultados para la estratificación de territorios.
- La ficha diagnóstico que se propone aporta información relevante para la toma de decisiones en materia de salud, a partir de la identificación de grupos poblacionales y territorios con mayor riesgo.
- La identificación de los requisitos permitió un mayor entendimiento de las necesidades del cliente.
- Se identificaron 7 HU divididas en 4 iteraciones y con la planificación del esfuerzo dedicado al desarrollo de cada una de ellas, se logró una mejor organización del trabajo y el establecimiento de fechas de culminación para cada iteración.
- La utilización del estilo arquitectónico en capas permitió una mejor estructuración de la aplicación.

CAPÍTULO 3: VERIFICACIÓN DE LA SOLUCIÓN

En el siguiente capítulo se realizan las tareas de ingenierías correspondientes a las HU previamente identificadas. Se realizan las pruebas definidas por la metodología seleccionada, como son, las pruebas unitarias para verificar el código, y las pruebas de aceptación para comprobar si al final de cada iteración se consiguió la funcionalidad requerida. Además se define el estándar de codificación a utilizar en el desarrollo de la solución. Se realiza un caso de estudio para verificar la validez de los resultados de la solución propuesta.

3.1 Fase de implementación

Luego de la etapa de planificación y diseño se pasa a la de codificación o implementación de la solución, donde se cumple el plan de iteraciones. En esta fase se implementan las HU seleccionadas por cada iteración y se crean las tareas de ingeniería para ayudar a organizar la implementación exitosa de las HU. Luego de esta fase el cliente y el desarrollador estarán listo para realizar las pruebas.

3.1.1 Tareas de ingeniería

Cada HU está compuesta por una o varias tareas de ingeniería, éstas se realizan para especificar las acciones llevadas a cabo por los programadores. En el presente trabajo se crearon un total de 10 tareas de ingeniería por HU, de las cuales 5 forman parte de la primera iteración, donde se implementan los métodos que determinan los territorios que más afectan al riesgo y el comportamiento de los indicadores dentro de los estratos. En la segunda iteración se implementa un mecanismo que importa la fuente de datos y realiza el cálculo de percepción, en la siguiente iteración se ejecutan las llamadas a los métodos implementados, y se integra la salida de estos con la vista que se visualizará en la cuarta iteración en conjunto con el mapa temático.

En la tabla 9 se detallan para la iteración número uno, las tareas a desarrollar por cada HU y en la tabla 10 se describe una tarea de ingeniería que responde a una HU arquitectónicamente significativa, el resto se encuentran especificadas en anexos.

Tabla 9. Distribución de tareas de ingeniería por HU (iteración 1).

HU	**Tareas de ingeniería por HU**
Obtener información de los territorios y estratos más afectados	• Implementar un método que obtenga los territorios que más aportan al riesgo de cada estrato y retorne un listado de estos territorios. • Implementar un método que obtenga los territorios con todos sus indicadores con un promedio mayor que el aporte al riesgo de salud de ese indicador.

Tabla 10. Tarea de Ingeniería Obtener indicadores que más inciden en el aporte de riesgo de salud de los territorios.

Tarea de ingeniería	
Número Tarea: 1	**Número Historia de Usuario:** HU # 1
• **Nombre Tarea:** Obtener indicadores que más inciden en el aporte de riesgo de salud de los territorios.	
Tipo Tarea: Desarrollo	**Puntos Estimados:** 2
Fecha Inicio: 16/02/2018	**Fecha Fin:** 20/02/2018
Programador Responsable: Dalber I. Morán González	
Descripción: Esta tarea permite obtener los territorios que más inciden en el riesgo de salud de los estratos.	

3.1.2 Estándares de codificación

XP resalta que la comunicación de los programadores es a través del código, por lo que es necesario que sigan ciertos estándares de programación para lograr un entendimiento entre los programadores, de manera que cualquier persona del equipo de desarrollo pueda modificar el código. Además, se hace preciso que el código sea entendible para que posteriormente otros programadores puedan apoyarse en ese trabajo y desarrollen otras soluciones.

En el caso de la herramienta que se desarrolla, el estándar que utiliza es:

Máxima longitud de las líneas

- Todas las líneas se limitan a un máximo de 79 caracteres.

Importaciones

- Las importaciones se encuentran en líneas separadas.

Comentarios

- Se utilizan comentarios de una línea para hacer más entendible el código.

Comentarios de una línea: comentario pequeño que solo abarca una línea y describe el código que le sigue.

```
# Esto es un comentario de una línea
```

Estilo de los nombres

- **Clases e Interfaces:** los nombres de las clases presentan la primera letra en mayúscula, en caso de ser un nombre compuesto, la inicial de cada palabra se representa en mayúscula. Se utilizan nombres simples y de alguna manera que describan el contenido, se usan palabras completas, a no ser que la abreviatura sea muy conocida.

- **Métodos y variables:** los nombres de los métodos se representan en minúscula, en caso de ser un nombre compuesto, la inicial de la primera palabra se simboliza en minúscula, y la de las otras palabras que lo componen en mayúscula. Los nombres de las variables son cortos pero con significados lógicos, capaces de permitir a un observador identificar su función.

3.2 Fase de Pruebas

Con el desarrollo de una solución informática existe la necesidad de realizarle una gran cantidad de pruebas con el objetivo de verificar que el código esté correcto. Estas pruebas por lo general tienen que ser ejecutadas en varias ocasiones y se ven afectadas por los cambios que se introducen conforme se va construyendo la solución. XP divide las pruebas en dos grupos: pruebas de aceptación, o pruebas funcionales diseñadas por el cliente final, destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida y pruebas unitarias, encargadas de verificar el código y diseñadas por los programadores.

3.2.1 Pruebas de aceptación

Las pruebas de aceptación XP son especificadas por el cliente, y se centran en las características y funcionalidades generales del sistema, que son visibles y revisables por parte del usuario. Estas pruebas derivan de las HU que se han implementado como parte de la liberación del software (Joskowicz 2008).

Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una HU no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. Dado que la responsabilidad es grupal, es recomendable publicar los resultados de las pruebas de aceptación, de manera que todo el equipo esté al tanto de esta información (Joskowicz 2008).

Casos de prueba

En las tablas 11 y 12 se muestran los casos de prueba de aceptación aplicados a las HU: “Obtener información de los territorios y estratos más afectados” y “Visualizar resultados en mapa temático”.

Tabla 11. Caso de prueba de aceptación. Obtener información de los territorios y estratos más afectados.

Caso de prueba de aceptación	
Código: HU1_P1	**Historia de Usuario:** 1
Nombre: Obtener información de los territorios y estratos más afectados.	
Descripción: Prueba para validar la funcionalidad. Obtener información de los territorios y estratos más afectados.	
Condiciones de ejecución: • El sistema debe haber obtenido la capa para los territorios a evaluar seleccionada por el usuario. • El sistema debe haber obtenido los datos de los indicadores estadísticos y/o los datos de los indicadores cartográficos. • El sistema debe haber obtenido la estratificación seleccionada por el usuario. • El usuario debe escoger la opción *Post-procesamiento*.	
Resultados esperados: En caso que se cumplan las condiciones de ejecución, el sistema obtiene información de los territorios y estratos más afectados, por tanto la salida del caso de prueba sería una lista de indicadores y territorios que afectan en cada estrato. En caso contrario el sistema muestra un mensaje informando el motivo por el cuál no realizó la acción.	
Evaluación de la prueba: Prueba satisfactoria	

Tabla 12. Caso de prueba de aceptación. Visualizar resultados en mapa temático.

Caso de prueba de aceptación	
Código: HU6_P1	**Historia de Usuario:** 6
Nombre: Visualizar resultados en mapa temático.	
Descripción: Prueba para validar la funcionalidad. Visualizar resultados en mapa temático.	
Condiciones de ejecución: • El sistema realiza el agrupamiento de los territorios por estratos. • El sistema muestra los resultados del agrupamiento en la cartografía. • El sistema muestra el valor de riesgo de salud de los estratos generados.	
Resultados esperados: En caso que se cumplan las condiciones de ejecución, el sistema muestra una vista con el mapa con los territorios divididos por estratos y el riesgo de salud de los estratos. En caso contrario el sistema muestra un mensaje informando el motivo por el cuál no realizó la acción.	
Evaluación de la prueba: Prueba satisfactoria	

Análisis de los resultados

Para validar que el resultado obtenido por el sistema coincide con el resultado esperado por el cliente se diseñaron un total de 7 casos de prueba de aceptación en conjunto cliente-desarrolladores. De este total, 2 arrojaron el resultado esperado mientras que 5 pruebas resultaron fallidas, las funcionalidades que respondían a estas pruebas fueron tratadas en la siguiente iteración, al volver a aplicar las pruebas de funcionalidad solo una resultó fallida luego se trataron sus funcionalidades y en la iteración el resultado fue exitoso. Finalmente se obtuvieron un total de 7 pruebas satisfactorias de 7 casos de prueba aplicados.

A continuación se muestran los resultados por iteración

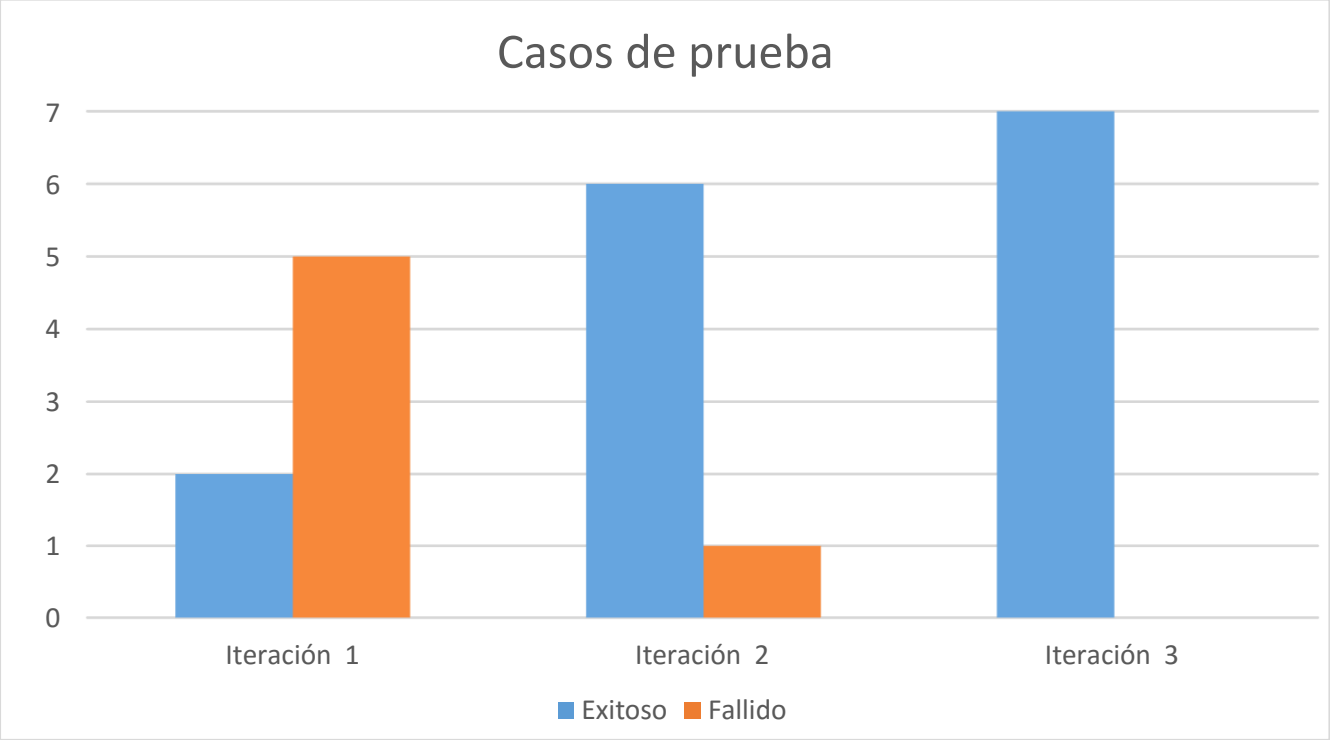

Figura 11. Resultado de aplicar la prueba de aceptación.

3.2.2 Pruebas de caja blanca

Las pruebas de caja blanca se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente. Se escogen distintos valores de entrada para examinar cada uno de los posibles flujos de ejecución del programa cerciorándose que se devuelvan los valores de salida adecuados (Pressman 2005).

Las pruebas de caja blanca intentan garantizar que:

- Se ejecutan al menos una vez todos los caminos independientes de cada módulo.
- Se utilizan las decisiones en su parte verdadera y en su parte falsa.
- Se ejecuten todos los bucles en sus límites.
- Se utilizan todas las estructuras de datos internos.

La técnica utilizada dentro de las pruebas de caja blanca fue camino básico. En la figura 11 se enumeran las sentencias de código del método *indXeMediaIndicador( )*.

```
def indEMediaInd(self):
    indicadores = {}
    mediaIndicadores = self.calcularMediasInd()
    for territorio,indicador in [(territorio,indicador)for territorio in self.territorios for indicador in territorio.indicadores]:
            if indicador.aporteRiesgo >= mediaIndicadores[indicador.nombre]:
                if territorio in indicadores:
                    indicadores[territorio].append(indicador)
                else:
                    indicadoresTmp =[]
                    indicadoresTmp.append(indicador)
                    indicadores[territorio] = indicadoresTmp
    return indicadores
```

Figura 12. Código del método indXeMediaIndicador( ).

Luego de haberse construido el grafo se realiza el cálculo de la complejidad ciclomática[15] mediante las tres fórmulas descritas a continuación, las cuales deben arrojar el mismo resultado para asegurar que el cálculo de la complejidad sea correcto.

1. La complejidad ciclomática coincide con el número de regiones del grafo de flujo.
2. La complejidad ciclomática, V(G), de un grafo de flujo G, se define como V(G) = Aristas - Nodos+2.
3. La complejidad ciclomática, V(G), de un grafo de flujo G, también se define como V(G) = Nodos de predicado[16] + 1.

A partir del grafo de flujo del método *indXeMediaIndicador( )* que se presenta en la figura 13, la complejidad ciclomática sería:

- Como el grafo tiene cuatro regiones, V(G) = 5
- Como el grafo tiene 11 aristas y 8 nodos, V(G) = 11- 8 + 2 = 5

[15] Complejidad ciclomática: es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa.
[16] Nodo predicado: es él que representa una condicional if o case, es decir, de él salen varios caminos.

- Como el grafo tiene 4 nodos de predicado, V(G) = 4 + 1 = 5

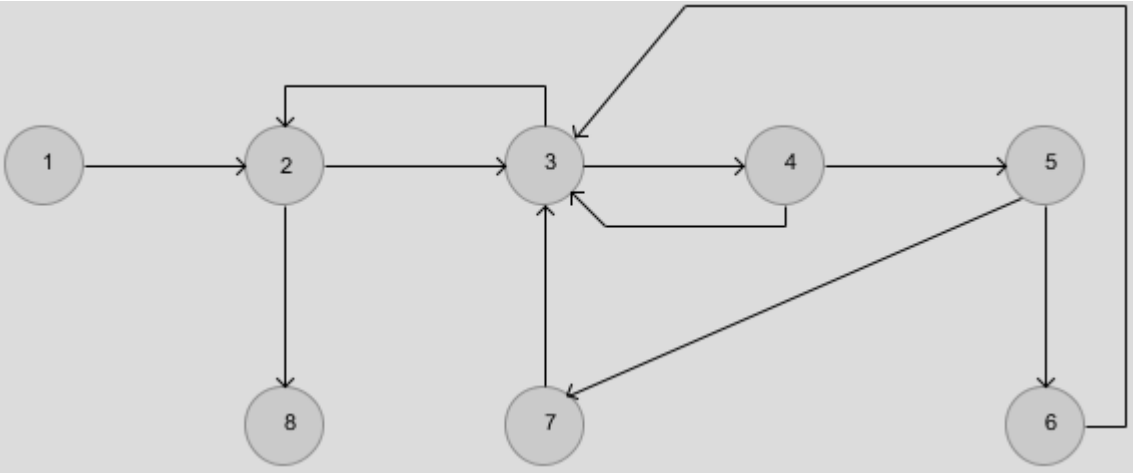

Figura 13. Grafo de flujo del método indXeMediaIndicador( ).

Debido a que el cálculo de las tres fórmulas anteriormente mencionadas arrojó el mismo resultado se puede plantear que la complejidad ciclomática del método es 5. Esto significa que existen 5 posibles caminos por donde el flujo puede circular. Este valor representa el número mínimo de casos de pruebas para el procedimiento tratado.

Caminos básicos identificados:

- Camino 1: 1-2-8
- Camino 2: 1-2-3-2-8
- Camino 3: 1-2-3-4-3-2-8
- Camino 4: 1-2-3-4-5-6-3-2-8
- Camino 5: 1-2-3-4-5-7-3-2-8

Para cada camino básico determinado se realiza un diseño de caso de prueba.

Tabla 13. Caso de prueba para el camino básico #1.

Caso de prueba para el camino básico #1 (1-2-8)	
Descripción	Prueba para comprobar los resultados de la función *indXeMediaIndicador()* en caso que la lista de territorios a evaluar sea vacía.
Condición de ejecución	• longitud de self.territorios = 0
Entrada	• self.territorios=[ ] • territorio.indicadores=[ ]
Resultado	indicadores = { }

Resultado de la prueba	Prueba satisfactoria

Tabla 14.Caso de prueba para el camino básico #2.

Caso de prueba para el camino básico #2 (1-2-3-2-8)	
Descripción	Prueba para validar los resultados de la función *indXeMediaIndicador()* en caso que la lista de indicadores sea vacía.
Condición de ejecución	• longitud de self.territorios > 0 • longitud de territorio.indicadores=0
Entrada	• self.territorios=[1,2,3,4,5] • territorio.indicadores=[ ]
Resultado	indicadores = { }
Resultado de la prueba	Prueba satisfactoria

Tabla 15. Caso de prueba para el camino básico #3

Caso de prueba para el camino básico #3 (1-2-3-4-3-2-8)	
Descripción	Prueba para validar los resultados de la función *indXeMediaIndicador()* en caso que el aporte de riesgo del indicador sea menor que el promedio de riesgo del mismo.
Condición de ejecución	• longitud de self.territorios > 0 • longitud de territorio.indicadores > 0 • indicador.aporteRiesgo < mediaIndicadores[indicador.nombre]
Entrada	• self.territorios=[1,2,3,4,5] • territorio.indicadores=[1,2,3,4]
Resultado	indicadores = { }
Resultado de la prueba	Prueba satisfactoria

Tabla 16. Caso de prueba para el camino básico #4

Caso de prueba para el camino básico #4 (1-2-3-4-5-6-3-2-8)	
Descripción	Prueba para validar los resultados de la función *indXeMediaIndicador()* en caso que el aporte de riesgo del indicador sea mayor que el promedio de riesgo del mismo y que se encuentre ese territorio en la lista de indicadores.
Condición de ejecución	• longitud de self.territorios > 0 • longitud de territorio.indicadores > 0 • indicador.aporteRiesgo >= mediaIndicadores[indicador.nombre] • if territorio in indicadores
Entrada	• self.territorios=[1,2,3,4,5] • territorio.indicadores=[1,2,3,4]

Resultado	indicadores = {1: 3}
Resultado de la prueba	Prueba satisfactoria

Tabla 17. Caso de prueba para el camino básico #5

Caso de prueba para el camino básico #5 (1-2-3-4-5-7-3-2-8)	
Descripción	Prueba para validar los resultados de la función *indXeMediaIndicador()* en caso que no se encuentre el territorio analizado, en la lista de indicadores.
Condición de ejecución	• longitud de self.territorios > 0 • longitud de territorio.indicadores > 0 • indicador.aporteRiesgo >= mediaIndicadores[indicador.nombre] • if not territorio in indicadores
Entrada	• self.territorios=[1,2,3,4,5] • territorio.indicadores=[1,2,3,4]
Resultado	Indicadores = {2: 3, 4: 4}
Resultado de la prueba	Prueba satisfactoria

3.3 Caso de estudio

Con el fin de valorar los resultados de la solución propuesta se decide aplicar a un caso de estudio donde se realiza un proceso de estratificación de las dieciséis provincias de Cuba definidas en la última división política-administrativa que se llevó a cabo. Se utilizó como fuente de información el Anuario Estadístico del año 2016 (INFOMED 2016) y se seleccionaron los indicadores siguientes:

- Enfermedades crónicas no transmisibles
- Mortalidad por enfermedades infecciosas intestinales
- Mortalidad perinatal
- Enfermedades del corazón
- Prevalencia de asma bronquial
- Cirrosis y otras enfermedades crónicas del hígado
- Enfermedades crónicas del as vías respiratorias inferiores
- Accidentes
- Diabetes mellitus
- Lesiones autoinflingidas internacionalmente
- Enfermedades transmisibles, causas de muerte materna, perinatal y nutricional
- Tumores malignos

- Influenza y neumonía
- Incidencia de lepra según clasificación operacional
- Enfermedades cerebrovasculares
- Notificaciones de tuberculosis y asma bronquial
- Enfermedades de las arterias, arteriolas y vasos capilares
- Mortalidad por enfermedades infecciones y parasitarias

Aplicación a un caso de estudio

Para realizar la clasificación de cada una de las provincias de Cuba utilizando la herramienta desarrollada se empleó el algoritmo de agrupamiento k-medias y el número de estratos se fijó en 4. La figura 15 muestra parte del proceso realizado.

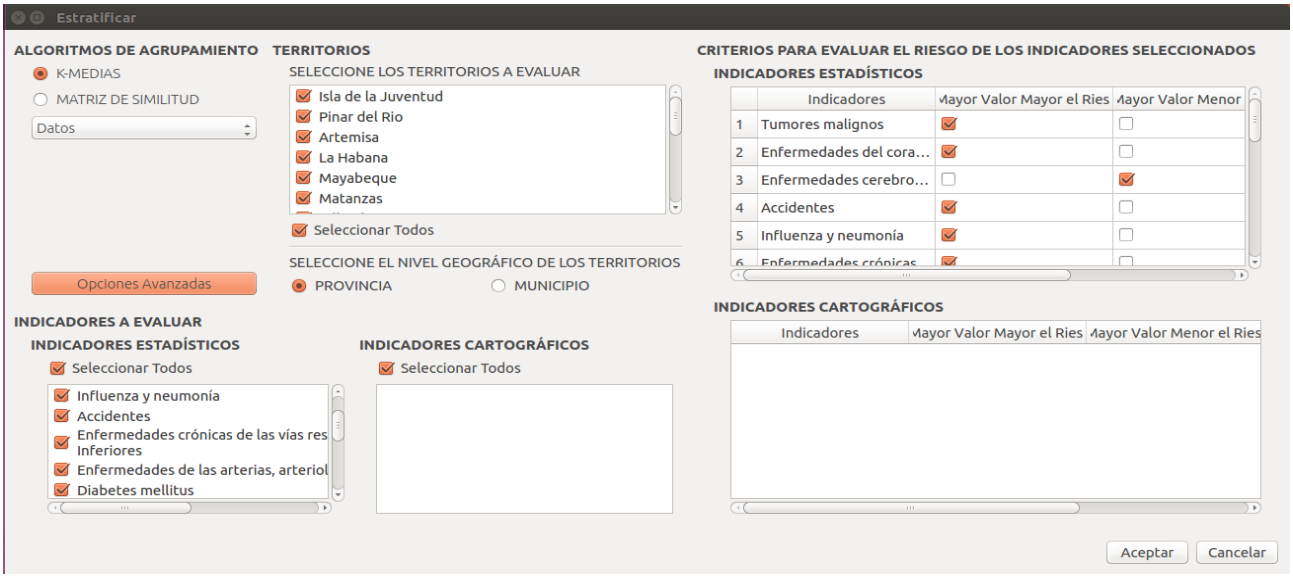

Figura 14. Interfaz de usuario VistaEstratificacion.

Resultados de la aplicación del caso de estudio

A continuación se muestran los resultados de la estratificación de las provincias de Cuba a partir del proceso analítico-estadístico de las variables de salud escogidas. En la Figura 16 se presentan en forma de mapa y en la tabla de manera más detallada. El mapa muestra los territorios de cada estrato en una escala de color que representa el nivel de riesgo, donde el color azul y el rojo constituyen el mínimo y máximo valor de riesgo respectivamente.

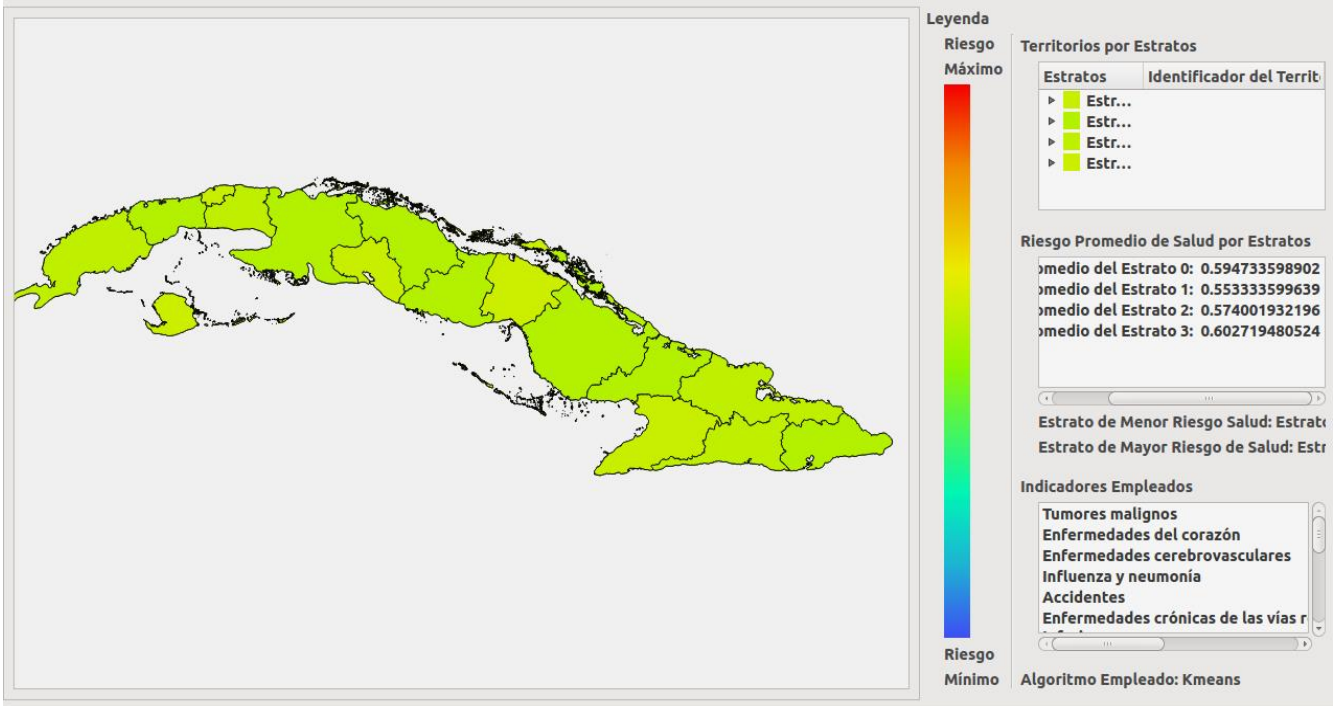

Figura 15. Mapa temático de la estratificación realizada utilizando la herramienta propuesta.

Tabla 18. Resultados de la estratificación realizada utilizando la herramienta propuesta.

Nombre del estrato	**Provincias**	**Riesgo de salud**
Estrato 0	La Habana, Cienfuegos, Granma	0.594733
Estrato 1	Artemisa, Matanzas, Las Tunas, Villa Clara, Sancti Spíritus, Camagüey, Guantánamo	0.553333
Estrato 2	Pinar del Río, Santiago de Cuba, Mayabeque, Holguín	0.574001
Estrato 3	Isla de la Juventud, Ciego de Ávila	0.602719

Luego del agrupamiento y la formación de los grupos, se realizó el post-procesamiento a los estratos. En la Figura 13 se muestra parte de este proceso realizado.

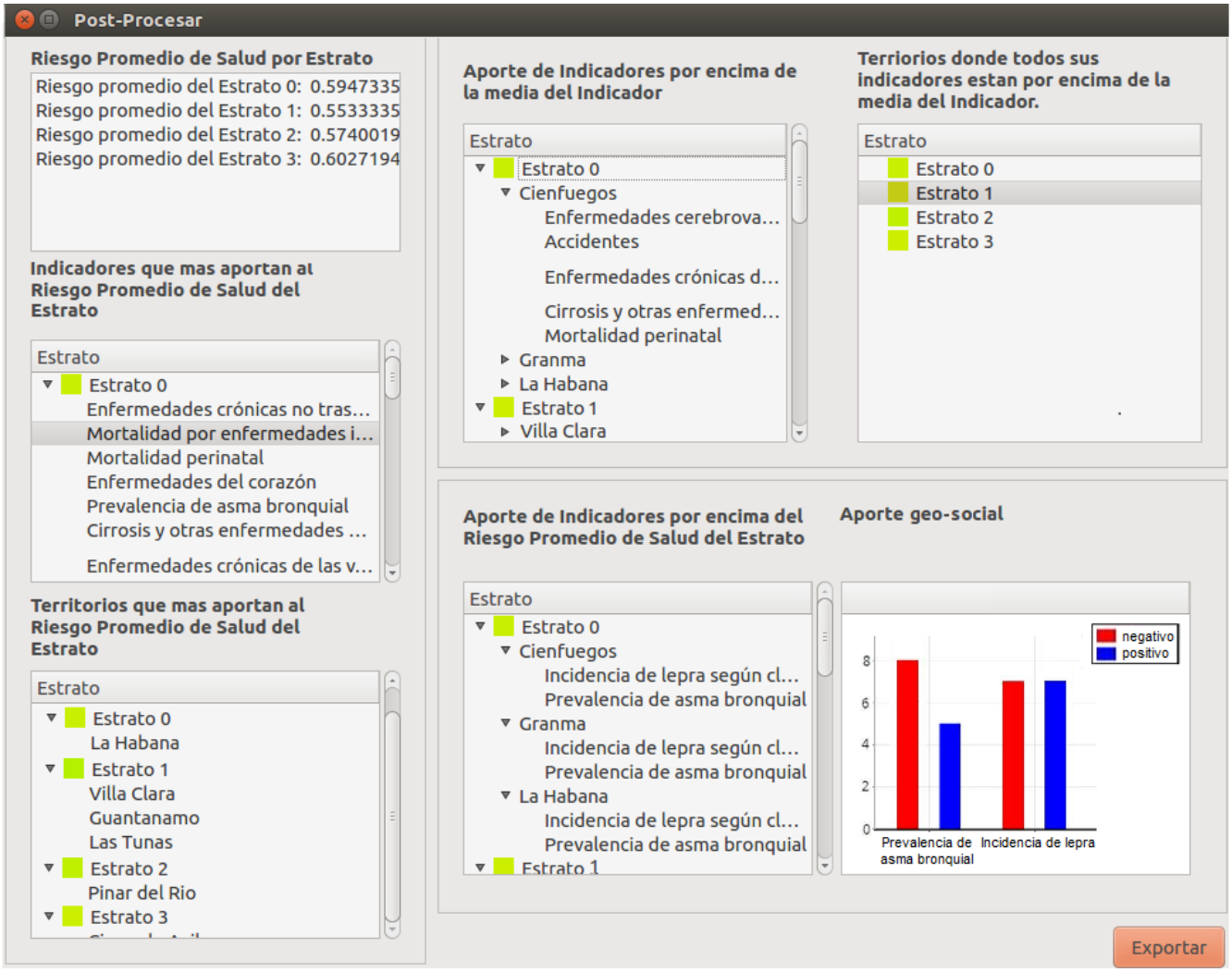

Figura 16. Interfaz de usuario VistaPost_procesamiento.

3.4 Resultados experimentales

Luego de realizarse la fase de post-procesamiento el algoritmo arrojó los resultados sobre los indicadores que afectan por estratos. Esta información sirve como base para realizar un diagnóstico sobre posibles factores que afectan en temas de salud. En la tabla 14 se muestra cada estrato formado mediante la etapa de prueba con los indicadores que más aportan al riesgo.

Tabla 19. Resultados del post-procesamiento de estratos utilizando la herramienta propuesta.

Estratos	Indicadores que afectan los estratos
Estrato 0	• Enfermedades crónicas no transmisibles • Mortalidad por enfermedades infecciosas intestinales • Mortalidad perinatal

	• Enfermedades del corazón • Prevalencia de asma bronquial • Cirrosis y otras enfermedades crónicas del hígado • Enfermedades crónicas del as vías respiratorias inferiores • Accidentes • Diabetes mellitus • Lesiones autoinflingidas internacionalmente • Enfermedades transmisibles, causas de muerte materna, perinatal y nutricional • Tumores malignos
Estrato 1	• Enfermedades crónicas no transmisibles • Mortalidad perinatal • Enfermedades del corazón • Prevalencia de asma bronquial • Cirrosis y otras enfermedades crónicas del hígado • Enfermedades crónicas del as vías respiratorias inferiores • Accidentes • Diabetes mellitus • Lesiones autoinflingidas internacionalmente • Enfermedades transmisibles, causas de muerte materna, perinatal y nutricional • Tumores malignos • Influenza y neumonía
Estrato 2	• Enfermedades crónicas no transmisibles • Mortalidad perinatal • Enfermedades del corazón • Prevalencia de asma bronquial • Cirrosis y otras enfermedades crónicas del hígado • Enfermedades crónicas del as vías respiratorias inferiores • Accidentes • Diabetes mellitus • Lesiones autoinflingidas internacionalmente • Enfermedades transmisibles, causas de muerte materna, perinatal y nutricional • Tumores malignos • Influenza y neumonía • Incidencia de lepra según clasificación operacional
Estrato 3	• Enfermedades crónicas no transmisibles • Mortalidad perinatal • Enfermedades del corazón • Prevalencia de asma bronquial • Cirrosis y otras enfermedades crónicas del hígado • Enfermedades crónicas del as vías respiratorias inferiores • Accidentes • Diabetes mellitus • Tumores malignos

Luego de mostrar los resultados por estratos de los indicadores que más aportan al riesgo, se muestra la siguiente ficha diagnóstico que representa el resultado de la investigación. La ficha diagnóstico está compuesta por los estratos y de cada uno de ellos: los datos obtenidos, que servirán para la toma de decisiones en la rama de la salud.

Tabla 20. Ficha diagnóstico.

Ficha diagnóstico		
Número de estrato: 0	Territorios : La Habana, Cienfuegos, Granma	
Riesgo de salud : 0.594733		
Territorios con mayor aporte al riesgo	La Habana	
Indicadores con mayor aporte al riesgo		**Percepción geo-social positiva**
• Enfermedades crónicas no transmisibles		72
• Mortalidad por enfermedades infecciosas intestinales		64
• Mortalidad perinatal		86

El análisis de los resultados de la aplicación a un caso de estudio, constituyó la base fundamental para la valoración de la solución. Luego de esta etapa se puede plantear que el algoritmo de post-procesamiento para la obtención de un diagnóstico sobre posibles factores de riesgos de salud:

- Constituye un mecanismo de gran aplicabilidad en la planificación territorial, debido a que permite valorar el potencial riesgo del territorio.
- Contribuye a identificar los denominados factores de riesgo.
- Identifica áreas geográficas susceptibles, con grupos poblacionales con mayor riesgo de enfermar o morir y por tanto aporta información también a la toma de decisiones encaminada a la atención diferenciada por áreas de salud.

Conclusiones parciales

- En el presente capítulo se detallaron las tareas de ingeniería correspondiente a cada HU, permitiendo la organización del trabajo en una secuencia lógica de pasos.
- Las pruebas de aceptación y de caja blanca efectuadas facilitaron detectar, documentar y corregir las no conformidades existentes en el sistema implementado.

- La realización del caso de estudio evidenció la efectividad de la solución presentada.
- La solución desarrollada posibilitó realizar análisis de los resultados obtenidos en el proceso de post-procesamiento en la estratificación realizada sobre las provincias de Cuba, identificando cuales de estas presentan mayor riesgo de salud.

CONCLUSIONES GENERALES

Como resultados de la presente investigación se obtuvo una propuesta de solución para el post-procesamiento de estratos en la estratificación de territorios utilizando SIG que contribuye al mejoramiento de la capacidad de gestión de las entidades de salud. En función de los resultados obtenidos se arribó a las siguientes conclusiones:

- La identificación de los territorios y estratos más afectados y la determinación de los factores asociados al comportamiento de los mismos, impactan en los resultados finales de la estratificación territorial y en la obtención de información para tomar decisiones sobre estudios estratificados.
- El algoritmo de post-procesamiento propuesto permite obtener los territorios y estratos más afectados en función de los indicadores de salud que inciden en el comportamiento de los mismos.
- La integración de la solución propuesta al componente de estratificación existente, facilitó la realización del proceso de post-procesamiento utilizando indicadores de naturaleza temática y espacial, esto facilita un diagnóstico sobre posibles factores de riesgo de salud por territorios en pos de brindar información para la toma de decisiones.
- Las pruebas aplicadas para la validación de la solución informática y la valoración de los resultados a través de un caso de estudio demostró que el sistema cumple con los requisitos definidos, garantizando su correcto funcionamiento.

RECOMENDACIONES

Incluir un algoritmo que obtenga los indicadores que más afectan en un estrato teniendo en cuenta la autocorrelación espacial [17].

[17] La autocorrelación espacial (AE): es la concentración o dispersión de los valores de una variable en un mapa.

REFERENCIAS BIBLIOGRÁFICAS

ABDULLAHI, S., SCHARDT, M. y PRETZSCH, H., 2017. An unsupervised two-stage clustering approach for forest structure classification based on X-band InSAR data—A case study in complex temperate forest stands. *International Journal of Applied Earth Observation and Geoinformation*, vol. 57, pp. 36–48.

ACOSTA, H.M.D., DÍAZ, S.M., BUERGO, D.R., GALINDO, M.V. y ACOSTA, M.M.S., 2013. Estratificación del bajo peso al nacer desde un enfoque de determinantes sociales. *Revista Finlay*, vol. 3, no. 1, pp. 40–50.

AGILE & SCRUM, 2016. Puntos de historia y velocidad en Scrum. *Management Plaza* [en línea]. [Consulta: 8 junio 2018]. Disponible en: http://managementplaza.es/blog/puntos-de-historia-velocidad-scrum/.

AGRAWAL, K.P., GARG, S., SHARMA, S. y PATEL, P., 2016. Development and validation of OPTICS based spatio-temporal clustering technique. *Information Sciences* [en línea], vol. 369, pp. 388–401. [Consulta: 5 noviembre 2016]. Disponible en: http://www.sciencedirect.com/science/article/pii/S0020025516304765.

ALEGRET RODRÍGUEZ, M., HERRERA, M. y GRAU ABALO, R., 2008. Las técnicas de estadística espacial en la investigación salubrista: caso síndrome de Down. *Revista Cubana de Salud Pública* [en línea], vol. 34, no. 4, pp. 0–0. [Consulta: 7 octubre 2015]. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0864-34662008000400003.

ARIADNA, C.M. y MARÍA DEL CARMEN, P.B., 2015. Clasificación del Territorio Nacional Según un Índice de Condiciones de Vida, Cuba 2014. *Convención Salud 2015* [en línea]. S.l.: s.n., [Consulta: 25 abril 2016]. Disponible en: http://www.convencionsalud2015.sld.cu/index.php/convencionsalud/2015/paper/viewPaper/596.

ARSANJANI, J.J., ZIPF, A., MOONEY, P. y HELBICH, M., 2015. *OpenStreetMap in GIScience: Experiences, Research, and Applications* [en línea]. S.l.: Springer. [Consulta: 9 febrero 2016]. Disponible en: https://www.google.com/books?hl=es&lr=&id=uHIKBwAAQBAJ&oi=fnd&pg=PR5&dq=giscience&ots=SZ09_Cu-pR&sig=SvM0Esc0r1ZFhFpqhz4vqsI8wZg.

ASENSIO BLASCO, E., 2014. Aplicación de técnicas de minería de datos en redes sociales/web. ,

AYALA, 2002. *Riesgos Naturales*. 2002. S.l.: s.n.

BATISTA MOLINER, R., FEAL CAÑIZARES, P., COUTIN MARIE, G., RODRÍGUEZ MILORD, D. y GONZÁLEZ CRUZ, R., 2001. *Guía para la realización del proceso de estratificación epidemiológica. La Habana: MINSAP. 2001.* 2001. S.l.: s.n.

BECK, K., 2000. Extreme programming explained: embrace change. *Addison-Wesley Professional* [en línea], Disponible en: http://books.google.es/books?hl=es&lr=&id=G8EL4H4vf7UC&oi=fnd&pg=PR13&dq=Extreme+Programing +Explained&ots=j9vFtsgXyl&sig=Xz6T5Ne01wTeLnPskTctYLBSTdo.

REFERENCIAS BIBLIOGRÁFICAS

BETANCOURT, Y.G.P., 2014. La minería de datos espaciales. Principales tendencias y perspectivas en los estudios salubristas. *GConocimiento*, vol. 5, no. 9.

BOSQUE SENDRA, J., 2013. Cartografía de riesgos naturales en América Central. ,

BRAVO, J.D., 2000. *Breve introducción a la cartografía ya los sistemas de información geográfica (SIG)* [en línea]. 2000. S.l.: s.n. [Consulta: 20 mayo 2018]. Disponible en: http://hc.rediris.es/pub/bscw.cgi/d251342/itcsig.pdf.

CANGREJO ALJURE, D. y AGUDELO, J.G., 2011. Minería de datos espaciales. ,

CASAS, S. y REINAGA, H., 2008. *Identificación y modelado de aspectos tempranos dirigido por tarjetas de responsabilidades y colaboraciones. In : XIV Congreso Argentino de Ciencias de la Computación [online]. 2008. [Accessed 21 May 2015]. Available from: http://sedici.unlp.edu.ar/handle/10915/21813*. S.l.: s.n.

CHENGFU, X., XIAOJUN, M. y YUJIAN, W., 2010. Perspectives on GIS Development in China. *Information Technology and Applications, International Forum on*, vol. 3, pp. 385–388. DOI http://doi.ieeecomputersociety.org/10.1109/IFITA.2010.180.

COBO, Á., 2007. Diseño y programación de bases de datos. [en línea], no. Editorial Visión Libros. Disponible en: http://books.google.es/books?hl=es&lr=&id=anCDr9NkGsC&oi=fnd&pg=PA7&dq=Dise%C3%B1o+y+programaci%C3%B3n+de+bases+de+datos&ots=UXEBp8mpzV&sig=jPWxCyBUit3XHIQIr4NpzhIbUwQ.

COX, J., SOVANNAROTH, S., SOLEY, L.D., NGOR, P., MELLOR, S. y ROCA-FELTRER, A., 2014. Novel approaches to risk stratification to support malaria elimination: an example from Cambodia. *Malaria journal*, vol. 13, no. 1, pp. 371.

CRAIG, L., 1999. *UML y Patrones. Introducción al análisis y diseño orientado a objetos. Pretince Hall, Hispanoamérica México. 1999*. 1999. S.l.: s.n.

CRUZ, O., 2011. Estratificación de riesgo de transmisión de Dengue en el municipio Playa utilizando el índice resumido para lograrlo, comparación de tres años de trabajo. ,

DELGADO ACOSTA, H., GONZÁLEZ MORENO, L., VALDÉS GÓMEZ, M., HERNÁNDEZ MALPICA, S., MONTENEGRO CALDERÓN, T. y RODRÍGUEZ BUERGO, D., 2015. Estratificación de riesgo de tuberculosis pulmonar en consejos populares del municipio Cienfuegos. *MediSur* [en línea], vol. 13, no. 2, pp. 275–284. [Consulta: 25 abril 2016]. Disponible en: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S1727-897X2015000200005.

DUQUE, R.G., 2011. Python para todos. [en línea], [Consulta: 11 junio 2018]. Disponible en: http://dspace.universia.net/handle/2024/919.

ELDAWY, A., MOKBEL, M.F. y OTHERS, 2016. The Era of Big Spatial Data: A Survey. *Foundations and Trends® in Databases* [en línea], vol. 6, no. 3-4, pp. 163–273. [Consulta: 1 febrero 2017]. Disponible en: http://ftp.nowpublishers.com/article/Details/DBS-054.

ENTORNOS DE PROGRAMACIÓN, 2012. *Entornos de programación* [en línea]. 2012. S.l.: s.n. [Consulta: 21 mayo 2018]. Disponible en: http://lml.ls.fi.upm.es/ep/entornos.html#toc5.

REFERENCIAS BIBLIOGRÁFICAS

ERIC RODRIGUEZ, 2010. Patrones gof. [en línea], [Consulta: 26 mayo 2018]. Disponible en: http://www.academia.edu/5903473/Patrones_gof.

ESTER, M., FROMMELT, A., KRIEGEL, H.-P. y SANDER, J., 2000. Spatial data mining: database primitives, algorithms and efficient DBMS support. *Data Mining and Knowledge Discovery*, vol. 4, no. 2-3, pp. 193–216.

FENG, C.-C., WANG, Y.-C. y CHEN, C.-Y., 2014. Combining Geo-SOM and Hierarchical Clustering to Explore Geospatial Data. *Transactions in GIS* [en línea], vol. 18, no. 1, pp. 125–146. [Consulta: 14 marzo 2017]. Disponible en: http://onlinelibrary.wiley.com/doi/10.1111/tgis.12025/full.

FRANCISCO JOSÉ CORTIJO BON, 2001. Tecnicas no supervisadas : Métodos de agrupamiento. [en línea]. [Consulta: 8 junio 2018]. Disponible en: https://www.google.com.cu/search?q=Tecnicas+no+supervisadas%20:+M%C3%A9todos+de+agrupamiento+Bon+Francisco+Cortijo&cad=h.

GAJEWSKI, B. y MARTYN, T., 2016. Spatial data clustering in independent mobile environment. *Measurement Automation Monitoring* [en línea], vol. 62. [Consulta: 22 diciembre 2016]. Disponible en: https://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-95117230-511c-45d0-b1ae-83d4cf2926d8.

GARCÍA PÉREZ, C. y ALFONSO AGUILAR, P., 2013. Estratificación epidemiológica de riesgo. , DOI 2 de diciembre de 2013.

GARRE, M. y CUADRADO, J.J., 2017. Comparación de diferentes algoritmos de clustering en la estimación de coste en el desarrollo de software. ,

GILBERT, K. y NONELL, R., 2005. Knowledge discovery with clustering: Impact of metrics and reporting phase by using klass. ,

GONZÁLEZ, D.P., 2010. Algoritmos de agrupamiento basados en densidad y variación de clusters. [en línea], [Consulta: 20 mayo 2015]. Disponible en: http://www.cerpamid.co.cu/sitio/files/DamarisTesis.pdf.

HAN, PEI, & KAMBER, 2011. 2011. S.l.: s.n.

HERMAWATI, R. y SITANGGANG, I.S., 2016. Web-Based Clustering Application Using Shiny Framework and DBSCAN Algorithm for Hotspots Data in Peatland in Sumatra. *Procedia Environmental Sciences* [en línea], vol. 33, pp. 317-323. [Consulta: 18 abril 2016]. ISSN 18780296. DOI 10.1016/j.proenv.2016.03.082. Disponible en: http://linkinghub.elsevier.com/retrieve/pii/S1878029616002474.

HERNÁNDEZ, RAMÍREZ , & FERRI, 2007. *Introducción a la Minería de Datos.* 2007. S.l.: s.n.

INFOMED, 2016. 2016.

JAIN, A.., MURTY, N. y FLYNN, P.., 1999. *Data clustering: a review. ACM computing surveys (CSUR)*. 1999. S.l.: s.n.

REFERENCIAS BIBLIOGRÁFICAS

JEON, M. y OH, B.-W., 2016. Analyzing Spatial Data Using Clustering Algorithm for Urban Planning. *Advanced Science and Technology Letters* [en línea], vol. 135, no. CES-CUBE 2016, pp. 112-114. [Consulta: 30 agosto 2016]. ISSN 2287-1233 ASTL. DOI 10.14257/astl.2016.135.28. Disponible en: http://onlinepresent.org/proceedings/vol135_2016/28.pdf.

JETBRAINS INC, 2014. Python IDE & Django IDE for Web developers : JetBrains PyCharm. [en línea], [Consulta: 20 mayo 2018]. Disponible en: https://www.jetbrains.com/pycharm/.

JIA, P., CHENG, X., XUE, H. y WANG, Y., 2017. Applications of geographic information systems (GIS) data and methods in obesity-related research. *Obesity reviews*, vol. 18, no. 4, pp. 400–411.

JOSÉ DE CALDAS, F., 2016. *algoritmos de agrupamiento*. 2016. S.l.: s.n.

JOSKOWICZ, J., 2008. Reglas y prácticas en eXtreme Programming. Universidad de Vigo. , pp. 22.

JUAN PELÁEZ, 2009. *Arquitectura basada en capas* [en línea]. 2009. S.l.: s.n. Disponible en: http://www.juanpelaez.com/geek-stuff/arquitectura/arquitectura-basada-en-capas/.

KAO, J.-H., CHAN, T.-C., LAI, F., LIN, B.-C., SUN, W.-Z., CHANG, K.-W., LEU, F.-Y. y LIN, J.-W., 2017. Spatial analysis and data mining techniques for identifying risk factors of Out-of-Hospital Cardiac Arrest. *International Journal of Information Management* [en línea], vol. 37, no. 1, pp. 1528–1538. [Consulta: 9 febrero 2017]. Disponible en: http://www.sciencedirect.com/science/article/pii/S0268401216300512.

LETELIER, P., 2006. Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [en línea], [Consulta: 25 mayo 2018]. Disponible en: http://www.cyta.com.ar/ta0502/b_v5n2a1.htm.

LI, S., LI, W. y QIU, J., 2017. A Novel Divisive Hierarchical Clustering Algorithm for Geospatial Analysis. *ISPRS International Journal of Geo-Information* [en línea], vol. 6, no. 1, pp. 30. [Consulta: 1 febrero 2017]. Disponible en: http://www.mdpi.com/2220-9964/6/1/30/htm.

LÓPEZ CAVIEDES, M.A., 2004. Herramienta para la estratificación de municipios en zonas de riesgo para la salud. [en línea], [Consulta: 20 mayo 2015]. Disponible en: http://dgsa.uaeh.edu.mx:8080/xmlui/handle/123456789/29.

LOUDEN, K., 2004. Lenguajes de programación: Principios y práctica. Cengage Learning Latin America. ,

MARTÍN, A.C. y BARROS, M. del C.P., 2015. Diseño de un Índice de Condiciones de Vida y clasificación del territorio nacional. *Revista Cubana de Medicina General Integral* [en línea], vol. 31, no. 3. [Consulta: 19 abril 2016]. ISSN 1561-3038. Disponible en: http://www.revmgi.sld.cu/index.php/mgi/article/view/63.

MCDONNELL, R., DE LA FUENTE ARAGÓN, M.V. y MCDONNELL, R., 2012. Minería de Datos Aplicada a la Gestión de la Información Urbanística Data Mining Applied to Urban Information Management. *6th International Conference on Industrial Engineering and Industrial Management* [en línea]. S.l.: s.n., pp. 1476–1483. [Consulta: 7 octubre 2015]. Disponible en: http://www.adingor.es/congresos/web/articulo/detalle/a/2310.

REFERENCIAS BIBLIOGRÁFICAS

MIGUEL ANGEL, G., 2012. HERRAMIENTAS CASE. [en línea], Disponible en: https://prezi.com/aad8mbta_vjb/herramientas-case/ MOLINA.

MORALES PÉREZ, R. y VEGA TORRES, Y., 2015. *Propuesta para la estratificación de territorios basada en indicadores de salud.* junio 2015. S.l.: s.n.

OLIVA SANTOS, R., MACIÁ PEREZ, F. y GAREA LLANO, E., 2011. Esbozo de un modelo de integración de datos, metadatos y conocimiento geográfico. *VII Congreso Internacional Geomática 2011*. S.l.: s.n.,

PASCUAL, D., PLA, F. y SÁNCHEZ, S., 2007. Algoritmos de agrupamiento. Método Informáticos Avanzados. [en línea], [Consulta: 20 mayo 2015]. Disponible en: http://marmota.dlsi.uji.es/WebBIB/papers/2007/1_Pascual-MIA-2007.pdf.

PEÑA SUAREZ, A., 2017. *Modelo para la Caracterización del Delito en la Ciudad de Bogotá, Aplicando Técnicas de Minería de Datos Espaciales.* 2017. S.l.: s.n.

PÉREZ BETANCOURT, Y., 2016. *Estratificación de territorios basada en indicadores de salud sobre el Sistema de Información Geográfica QGIS.* 2016. S.l.: s.n.

PÉREZ BETANCOURT, Y., BETANCOURT, Y.G.P., POLANCO, L.G., PÉREZ, R.M. y VEGA, Y.T., 2016. Estratificación de territorios basada en indicadores de salud sobre el Sistema de Información Geográfica QGIS. *Revista Cubana de Ciencias Informáticas* [en línea], vol. 10, no. 0, pp. 163–175. ISSN 1994-1536. Disponible en: http://rcci.uci.cu/?journal=rcci&page=article&op=view&path[]=1374.

PÉREZ BETANCOURT, Y. y GONZÁLEZ POLANCO, L., 2013. La minería de datos espaciales y su aplicación en los estudios de salud y epidemiología. ,

PÉREZ BETANCOURT, Y., GONZÁLEZ POLANCO, L., MORALES PÉREZ, R. y VEGA TORRES, Y., 2016. Estratificación de territorios basada en indicadores de salud sobre el Sistema de Información Geográfica QGIS. ,

PÉREZ BETANCOURT, Y.G., 2018. Propuestas para el análisis geoespacial en estudios salubristas. [en línea]. [Consulta: 9 junio 2018]. Disponible en: https://www.google.com/search?source=hp&ei=_0wbW9v4Conf5gK_uofADA&q=infomed+cuba&oq=infome&gs_l=psy-ab.3.0.35i39k1j0l4j0i10k1j0l4.1438.2852.0.7140.7.6.0.0.0.0.170.967.0j6.6.0....0...1c.1.64.psy-ab..1.6.965.0..0i131k1.0.XGJ8l1qAcaM.

PÉREZ BETANCOURT, Y.G., GONZÁLEZ POLANCO, L. y FEBLES RODRÍGUEZ, J.., 2017. ESTRATIFICACIÓN DE TERRITORIOS BASADA EN INDICADORES DE SALUD Y MEDIDAS DE SIMILITUD GEOMÉTRICAS. [en línea]. [Consulta: 12 junio 2018]. Disponible en: https://scholar.google.com.cu/citations?user=O3auRnEAAAAJ&hl=es#d=gs_md_cita-d&p=&u=%2Fcitations%3Fview_op%3Dview_citation%26hl%3Des%26user%3DO3auRnEAAAAJ%26citation_for_view%3DO3auRnEAAAAJ%3ASe3iqnhoufwC%26tzom%3D240.

PERUMAL, M., VELUMANI, B., SADHASIVAM, A. y RAMASWAMY, K., 2015. Spatial Data Mining Approaches for GIS–A Brief Review. *Emerging ICT for Bridging the Future-Proceedings of the 49th Annual Convention of the Computer Society of India CSI Volume 2* [en línea]. S.l.: Springer,

pp. 579–592. [Consulta: 12 diciembre 2015]. Disponible en: http://link.springer.com/10.1007/978-3-319-13731-5_63.

POSTGIS DEVELOPMENT TEAM, 2014. PostGIS. [en línea], [Consulta: 25 mayo 2018]. Disponible en: http://postgis.refractions.net/.

POSTGRESQL-3 GLOBAL DEVELOPMENT GROUP, 2014. PostgreSQL: Documentation: 9.0: Release 9.0.1. [en línea], [Consulta: 20 mayo 2018]. Disponible en: http://www.postgresql.org/docs/9.0/static/release-9-0-1.html.

PRESSMAN, R., 2005. *Ingeniería del software. Un enfoque práctico. Sexta edición. Editoria l McGraw-Hill. Interamericana Editores, SA de CV México*. S.l.: s.n.

QGIS DEVELOPMENT TEAM, 2012. QGIS project! QGIS A Free and Open Source Geographic Information System. [en línea], [Consulta: 19 mayo 2018]. Disponible en: http://www.qgis.org/en/site/.

RIQUELME, J.., RUIZ, R. y GILBERT, K., 2006. *Mineria de Datos: Conceptos y tendencias Inteligencia artificial. Iberoamericana de Inteligencia Artificial.* 2006. S.l.: s.n. 11 - 18

ROBINSON, C., 2011. Basic introduction into pgAdmin III and SQL queries. [en línea], Disponible en: http://library.thehumanjourney.net/658/.

RODRÍGUEZ, J.E.R., BLANCO, E.A.R. y CAMACHO, R.O.F., 2013. *Clasificación de datos usando el método k-nn.* 2013. S.l.: s.n.

SCHEFER-WENZL, SIGRID, SOBERNIG, STEFAN y STREMBECK, MARK, 2013. Evaluating A Uml-Based Modeling Framework For Process-Related Security Properties: A Qualitative Multi-Method Study. In : ECIS. [en línea], pp. 134. [Consulta: 19 mayo 2018]. Disponible en: http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1357&context=ecis2013_cr.

SHEKHAR, S., CHAWLA, S., RAVADA, S., FETTERER, A., LIU, X. y LU, C., 1999. Spatial Databases-Accomplishments and Research Needs. *IEEE Transactions on Knowledge and Data Engineering*, vol. 11, pp. 45–55. ISSN 1041-4347. DOI http://doi.ieeecomputersociety.org/10.1109/69.755614.

SHI, X. y KWAN, M.-P., 2015. Introduction: geospatial health research and GIS. *Annals of GIS* [en línea], vol. 21, no. 2, pp. 93–95. [Consulta: 6 octubre 2015]. DOI 10.1080/19475683.2015.1031204. Disponible en: http://www.tandfonline.com/doi/full/10.1080/19475683.2015.1031204.

SOMMERVILLE, L., 2005. *Ingeniería del software* [en línea]. Pearson Educación. S.l.: s.n. [Consulta: 20 mayo 2018]. Disponible en: http://books.google.es/books?hl=es&lr=&id=gQWd49zSut4C&oi=fnd&pg=PA1&dq=Ingenieria+de+Softwar e+Ian+Sommerville&ots=s623rrszwd&sig=DVAJiDo0wOzveatAqj48nPXvtsI.

STARTED, G., 2010. *Software Design Tools for Agile Teams, with UML, BPMN and More* [en línea]. 2010. S.l.: s.n. Disponible en: http://www.visual-paradigm.com/.

SUN, W., XIANG, L., LIU, X. y ZHAO, D., 2016. An Improved K-medoids Clustering Algorithm Based on a Grid Cell Graph Realized by the P System. *Human Centered Computing* [en línea].

S.l.: Springer, pp. 365–374. [Consulta: 13 mayo 2016]. Disponible en: http://link.springer.com/chapter/10.1007/978-3-319-31854-7_33.

TAHA, A., 2016. Knowledge Discovery In GIS Data. *arXiv preprint arXiv:1601.07241* [en línea], [Consulta: 8 noviembre 2016]. Disponible en: http://arxiv.org/abs/1601.07241.

TAN, P., STEINBACH, M. y KUMAR, V., 2006. Classification: basic concepts, decision trees, and model evaluation. Introduction to data mining. ,

TANSER, F.C. y LE SUEUR, D., 2015. The application of geographical information systems to important public health problems in Africa. *International Journal of Health Geographics* [en línea], vol. 1, no. 4, pp. 9. [Consulta: 6 octubre 2015]. Disponible en: http://www.popline.org/node/233770.

TIAN, F., GAO, B., CUI, Q., CHEN, E. y LIU, T.-Y., 2014. Learning deep representations for graph clustering. *AAAI*. S.l.: s.n., pp. 1293–1299.

TSAI, C.-F. y CHIANG, Y., 2016. Enhancement of data clustering using TSS-DBSCAN approach for data mining. *Machine Learning and Cybernetics (ICMLC), 2016 International Conference on*. S.l.: IEEE, pp. 535–540.

VÍCTOR, O., 2011. *Sistemas de Información Geográfica. Libro SIG [online]* [en línea]. S.l.: s.n. Disponible en: http://wiki.osgeo.org/wiki/Libro_SIG.

WANG, J., DENG, Z., CHOI, K.-S., JIANG, Y., LUO, X., CHUNG, F.-L. y WANG, S., 2016. Distance metric learning for soft subspace clustering in composite kernel space. *Pattern Recognition*, vol. 52, pp. 113–134.

WEIIS, S. y INDURKHYA, N., 1998. *Predictive data mining: a practical guide. San Francisco, California: Morgan Kaufmann.* 1998. S.l.: s.n.

YASOBANT, S., VORA, K.S., HUGHES, C., UPADHYAY, A. y MAVALANKAR, D.V., 2015. Geovisualization: A Newer GIS Technology for Implementation Research in Health. *Journal of Geographic Information System* [en línea], vol. 7, no. 01, pp. 20. [Consulta: 6 octubre 2015]. Disponible en: http://www.scirp.org/journal/PaperInformation.aspx?paperID=53721.

YEE LEUNG, 2016. *Knowledge Discovery in Spatial Data*. 2016. S.l.: s.n.

YENISEI BOMBINO COMPANIONI, 2005. metodología de análisis para la estratificación según indicadores de salud. [en línea], [Consulta: 20 mayo 2018]. Disponible en: http://bvs.sld.cu/uats/rtv_files/2005/bombino.htm.