

**Universidad de las Ciencias Informáticas**

**Facultad 3**



***“Propuesta para la estratificación de territorios basada en indicadores de salud.”***

***Trabajo de Diploma para optar por el Título de Ingeniero en Ciencias Informáticas.***

**Autores:**

Yanislav Torres Vega.

Rolando Morales Pérez.

**Tutores:**

Ing. Yadian Guillermo Pérez Betancourt.

Ing. Liset González Polanco.

**Co-Tutor:**

Ing. Jorge Jesús Hidalgo Ruíz.

***“Ciudad de La Habana. Junio 2015”***

***“Año 57 de la Revolución”***



*“La magnitud de lo que logramos, no depende de lo que tenemos, sino de lo que seamos capaces de hacer.”*

*Che*

## **DECLARACIÓN DE AUTORÍA**

Declaro que somos los únicos autores del presente trabajo que lleva como título: “Propuesta para la estratificación de territorios basada en indicadores de salud” y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos el presente trabajo a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

\_\_\_\_\_

Yanislav Torres Vega

Autor

\_\_\_\_\_

Rolando Morales Pérez

Autor

\_\_\_\_\_

Ing. Yadian Guillermo Pérez Betancourt

Tutor

\_\_\_\_\_

Ing. Liset González Polanco

Tutor

## AGRADECIMIENTOS

*A mi mami por su dedicación, por todo su amor, por ser mi guía, por tu apoyo incondicional, por toda la confianza depositada en mí, no tengo palabras para expresar lo agradecida y orgullosa que me siento.*

*A mi hermanita Jeny, por ser más que una hermana, una amiga, mi razón de seguir y ser mejor para que sigas mi ejemplo, gracias por todos los momentos geniales que me has regalado, te quiero muchote.*

*A mi súper Lita por ser tan especial, por todo el cariño, confianza y por ayudarme a ser quien soy hoy.*

*A mi papi lindo, por todo tu amor, cariño, consejos, por confiar siempre en mí.*

*A mi Lito que aunque ya no este, siempre voy a estar agradecida de todo el amor y el cariño que me brindó, por ser mi segundo padre y apoyarme en todo momento hasta el final.*

*A mi tío Héctor por ser mi otro papá, por apoyarme en todo momento, por ser un ejemplo, por creer siempre en mí, y darme fuerzas cuando más la necesite, sin tu apoyo este sueño no hubiera sido posible.*

*A mis hermanitos Ely y Ale por ser tan dulces conmigo aunque a veces casi ni nos podíamos ver.*

*A mi tía Yuci, por ayudarme a realizar este sueño, por confiar siempre en mí, por brindarme tu apoyo y ser de cierto modo mi*  
*Ada madrina.*

*A mi amiga Roxi, por apoyarme siempre, por ser más que una amiga, una hermana, por tantos momentos geniales.*

*A mi amiga y mayor apoyo durante estos 5 años, Yanay, gracias por todos los consejos, por guiarme, por tu paciencia, y por*  
*tu cariño.*

*A todas mis amistades de la Universidad que durante 5 años se convirtieron en mi familia y principal apoyo, y lo seguirán siendo; resulta realmente imposible listarlos a todos y agradecerles con la magnitud que se merecen, muchas gracias a todos por los momentos vividos, y por hacer de esta etapa de la vida inolvidable.*

**Yany**

*Quiero agradecer a mi familia, en especial a mis padres por toda la confianza depositada en mí y por todo el apoyo que me han dado siempre en los momentos más difíciles de mi carrera.*

*A mi hermano Duniesky, que siempre ha sido mi ejemplo a seguir por su brillante inteligencia y dedicación al estudio.*

*A mi novia Jessica, por su cariño, apoyo y consejos, gracias por acompañarme en momentos difíciles y ser la motivación para seguir adelante.*

*A todos mis amigos por los momentos tan lindos que hemos compartido, en especial a Randy y a Roly por ayudarme en múltiples situaciones.*

**Roly**

**DEDICATORIA**

*A mi mami bella, por tu consagración, tu amor incondicional, por no haber perdido la confianza en mí, y ser mi ejemplo a seguir.*

*A mi linda Tata, por ser más que mi hermana, para que te sirva de ejemplo y veas que no hay cosas imposibles.*

*A mi Lita, por mostrarme el camino correcto, por ser mi apoyo, a la cual no tengo palabras para describir lo agradecida que estoy.*

*Y muy especial a mi Lito Manuel que aunque ya no se encuentre entre nosotros sé que le hubiese gustado acompañarme en este momento.*

**Yany**

*A mi mamá Miriam.*

*A mi papá Rofy.*

*A mi hermano Duniesky.*

*A todos mis abuelos.*

**Rofy**

## **RESUMEN**

El desarrollo de las tecnologías de la información y las comunicaciones en la actualidad ha potenciado el uso de la información geográfica en disímiles ramas como la agricultura, la meteorología, el turismo y la medicina, implicando la utilización de los mapas como factor importante en la toma de decisiones. El uso de los Sistemas de Información Geográfica (SIG) en la rama de la salud ha aumentado considerablemente, sin embargo su utilización en el sector se limita en gran medida a presentar información en la cartografía, sin explotar en su totalidad la componente espacial de los datos.

El objetivo del presente trabajo es desarrollar una propuesta para la estratificación de territorios basada en indicadores de salud. Para ello se efectuó un estudio del panorama actual de los softwares SIG y 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 por la metodología de desarrollo seleccionada. Se aplicó un caso de estudio para valorar los resultados de la solución propuesta.

Se obtuvo como resultado un sistema que permite integrar datos de variada naturaleza para el análisis y construcción de estratos; la solución tiene su base en las técnicas de agrupamiento de datos 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, estratificación, información geográfica, patrones de diseño, pruebas de software, sistemas de información geográfica, riesgo de salud, salud.

**ÍNDICE**

INTRODUCCIÓN ..... 1

CAPÍTULO 1: Referentes teóricos sobre la estratificación de territorios..... 5

    1.1 Estratificación ..... 5

        1.1.1 Estratificación territorial en temas de salud ..... 6

    1.2 Técnicas de agrupamiento ..... 7

        1.2.1 Algoritmos jerárquicos..... 7

        1.2.2 Algoritmos particionales ..... 8

        1.2.3 Algoritmos basados en densidad ..... 8

        1.2.4 Algoritmos basados en grafos..... 8

    1.3 Sistemas de Información Geográfica como soporte para el proceso de estratificación ..... 9

        1.3.1 Conceptos básicos asociados a los Sistemas de Información Geográfica ..... 10

        1.3.2 Panorama actual de aplicaciones SIG..... 11

    1.4 Análisis de soluciones existentes con soporte para la estratificación..... 12

    1.5 Herramientas, lenguajes y tecnologías a utilizar ..... 15

        1.5.1 Lenguaje de modelado..... 15

        1.5.2 Herramienta CASE..... 16

        1.5.3 Lenguaje de programación..... 17



1.5.4 Entorno de desarrollo integrado .....	18
1.5.5 Gestor de base de datos .....	19
1.6 Metodología de desarrollo .....	20
1.6.1 Programación extrema.....	21
Conclusiones parciales.....	22
CAPÍTULO 2: Propuesta de solución para la estratificación de territorios .....	24
2.1 Propuesta del proceso de estratificación .....	24
2.1.1 Pre-procesamiento de los datos.....	25
2.1.2 Agrupamiento de los datos.....	28
2.2 Requisitos de software .....	31
2.2.1 Requisitos funcionales .....	31
2.2.2 Requisitos no funcionales .....	32
2.3 Fase de planificación.....	32
2.3.1 Historias de usuarios.....	33
2.3.2 Estimación de esfuerzos por historias de usuario .....	34
2.3.3 Plan de iteraciones.....	35
2.3.4 Plan de entrega.....	35
2.4 Fase de diseño.....	36

2.4.1 Tarjetas Clase-Responsabilidad-Colaboración.....	36
2.5 Arquitectura de software.....	37
2.5.1 Estilo arquitectónico a utilizar.....	37
2.6 Patrones de diseño.....	38
2.6.1 Patrones Generales de Software para la Asignación de Responsabilidades.....	39
2.6.2 Patrones del Grupo de Cuatro .....	41
2.7 Diagrama de clases del diseño.....	42
2.8 Diagrama Entidad-Relación.....	43
Conclusiones parciales.....	43
CAPÍTULO 3: Implementación y validación de la solución.....	44
3.1 Fase de implementación.....	44
3.1.1 Tareas de ingeniería .....	44
3.1.1 Estándares de codificación .....	45
3.2 Fase de pruebas .....	46
3.2.1 Pruebas de aceptación .....	46
3.2.2 Pruebas de caja blanca.....	49
3.3 Caso de estudio.....	53
Conclusiones parciales.....	57

CONCLUSIONES GENERALES.....	58
RECOMENDACIONES .....	59
REFERENCIAS BIBLIOGRÁFICAS .....	60
GLOSARIO DE TÉRMINOS .....	64
ANEXOS.....	65
Anexo1. Historias de Usuario .....	65
Anexo2. Tareas de ingeniería asociadas a la HU # 3.1. ....	68
Anexo3. Tarjetas CRC.....	70
Anexo4. Casos de prueba de aceptación .....	73

**ÍNDICE DE FIGURAS**

Figura 1. Modelo conceptual de la propuesta de solución.....24

Figura 2. Evidencia de la arquitectura en capas.....38

Figura 3. Evidencia del patrón Experto. ....39

Figura 4. Evidencia del patrón Creador.....40

Figura 5. Evidencia del patrón Controlador. ....41

Figura 6. Evidencia del patrón Plantilla. ....42

Figura 7. Diagrama de clases. ....42

Figura 8. Diagrama Entidad-Relación. ....43

Figura 9. Código del método runCR( ). ....49

Figura 10. Grafo de flujo del método runCR( ). ....50

Figura 11. Interfaz de usuario VistaEstratificacion.....54

Figura 12. Mapa temático de la estratificación realizada utilizando la herramienta propuesta.....55

**ÍNDICE DE TABLAS**

Tabla 1. Comparación entre los sistemas analizados. .... 15

Tabla 2. Historia de usuario: Construir estratos. .... 33

Tabla 3. Historia de usuario: Exportar estratificación hacia una hoja de cálculo..... 34

Tabla 4. Estimación de esfuerzos por Historia de Usuario. .... 34

Tabla 5. Plan de duración de las iteraciones..... 35

Tabla 6. Plan de duración de las entregas. .... 35

Tabla 7. Tarjeta CRC para la clase Territorio. .... 36

Tabla 8. Tarjeta CRC para la clase Estratificación. .... 37

Tabla 9. Distribución de tareas de ingeniería por HU (iteración 1). .... 44

Tabla 10. Tarea de Ingeniería Agrupar territorios utilizando el algoritmo k-medias. .... 45

Tabla 11. Caso de prueba de aceptación Obtener características cartográficas a través de QGIS..... 47

Tabla 12. Caso de prueba de aceptación Construir estrato..... 48

Tabla 13. Caso de prueba para el camino básico #1. .... 51

Tabla 14. Caso de prueba para el camino básico #2. .... 51

Tabla 15. Caso de prueba para el camino básico #3. .... 52

Tabla 16. Caso de prueba para el camino básico #4. .... 52

Tabla 17. Resultados de la estratificación realizada utilizando la herramienta propuesta..... 55

Tabla 18. Tabla comparativa de los resultados de los procesos de estratificación realizados (por estratos)..... 56

## **INTRODUCCIÓN**

En la actualidad, el progresivo avance de las tecnologías computacionales y la necesidad del manejo de información periódicamente en aumento, conlleva a la necesidad de un mayor uso de los sistemas informáticos en diversas actividades de la sociedad. La agricultura, la meteorología, el turismo y la medicina, son algunas de las tantas ramas de aplicaciones de una herramienta informática muy popular y de gran impacto en los últimos tiempos: los Sistema de Información Geográfica (SIG) (Víctor 2011).

Para justificar su importancia y el papel que estos juegan hoy en día, es necesario citar el hecho de que aproximadamente un 80 % de la información que se maneja en cualquier tipo de disciplina se le puede asignar una posición geográfica (Víctor 2011). La situación es en la actualidad más favorable que nunca para el desarrollo de herramientas que permitan la utilización de toda esa información al tiempo que se consideran los datos relativos a su posición en el espacio.

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 (Rojas 1998). 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.

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.

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, Feal Cañizares, Coutin Marie, Rodríguez Milord, González Cruz 2001).

Actualmente la utilización de los SIG en el análisis de la distribución espacial de enfermedades ha aumentado considerablemente, esto se evidencia por las herramientas de análisis existentes que posibilitan la aplicación de una adecuada metodología de estratificación (Víctor 2011). Sin embargo estas no son extensibles, su utilización se limita a llevar información a la cartografía, y la componente espacial de los datos con que se trabaja no es explotada en su totalidad. Si bien el espacio es un elemento importante en los estudios salubristas, socio-económicos, demográficos y geopolíticos, no siempre se le da la importancia requerida, motivado por: i) acceso limitado a los SIG por los costos que ellos implican, ii) poco conocimiento de las herramientas y iii) el tiempo de formación en el área de los SIG es elevado.

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), sin embargo los métodos aplicados para la misma son puramente estadísticos y no se utilizan otras técnicas que tienen en cuenta la naturaleza espacial de los datos. 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), lo que reduce la eficiencia del trabajo. Estos elementos afectan 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 de salud.

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

¿Cómo realizar el proceso de estratificación de territorios para contribuir al mejoramiento de la capacidad de gestión de las entidades de salud?

Se plantea como **objeto de estudio** el proceso de estratificación de territorios centrando su **campo de acción** en la estratificación de territorios basado en indicadores de salud.

Para darle solución al anterior problema se plantea como **objetivo general** diseñar una propuesta de solución para la estratificación de territorios utilizando SIG que contribuya al mejoramiento de la capacidad de gestión de las entidades de salud.

**Objetivos específicos:**

1. Construir el marco teórico referencial de la investigación, relacionada con el proceso de estratificación de territorios basado en indicadores de salud.
2. Diseñar una propuesta de solución que facilite el proceso de estratificación de territorios utilizando Sistemas de Información Geográfica.
3. Aplicar la propuesta de solución en la implementación de una herramienta informática que permita estratificar territorios.
4. Verificar la solución informática propuesta aplicando diferentes pruebas y métricas.

Surge la siguiente ***idea a defender***:

Si se diseña una solución para la estratificación de territorios utilizando SIG entonces se contribuirá al mejoramiento de la gestión de las entidades de salud.

En el desarrollo del presente trabajo de investigación fueron utilizados los ***métodos científicos*** siguientes:

- **Histórico-Lógico:** Permitió realizar un estudio de las principales herramientas que realizan procesos de estratificación de territorios.
- **Análisis y Síntesis:** Se utiliza para identificar y analizar las diversas funcionalidades de los SIG que pueden ser aplicadas al proceso de estratificación territorial y su posterior síntesis, conforme a las necesidades de Cuba en el sector de la salud.
- **Modelación:** Se emplea para mostrar los diversos diagramas que se construyen como resultado del proceso de ingeniería de software.

**Posibles resultados:**

Obtener una propuesta de solución para la estratificación de territorios utilizando SIG que contribuya al mejoramiento de la capacidad de gestión de las entidades cubanas de salud.

**Validación:**

- Pruebas de aceptación para la solución informática.
- Aplicación de la propuesta solución en un caso de estudio.



## **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: Referentes teóricos sobre la estratificación de territorios.**

En este capítulo se presentan elementos teóricos relacionados con el proceso de estratificación para lograr un mejor entendimiento del trabajo a desarrollar. Se hace una valoración comparativa sobre varios softwares SIG existentes y sobre las principales herramientas que soportan procesos de estratificación. Además, se realiza un estudio de la metodología, herramientas, tecnologías y lenguajes a utilizar en el desarrollo de la solución.

### **Capítulo 2: Propuesta de solución para la estratificación de territorios.**

En este capítulo se realiza una descripción general 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: Implementación y validación 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: Referentes teóricos sobre la estratificación de territorios**

En este capítulo se presentan un conjunto de elementos teóricos que conforman el marco referencial relacionado con el objeto de estudio. Se realiza un estudio del panorama actual de los softwares SIG. Se realiza un análisis crítico de las herramientas existentes que soportan los procesos de estratificación de territorios. 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, Feal Cañizares, Coutin Marie, Rodríguez Milord, González Cruz 2001) .

Podrían citarse otras definiciones, pero 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:**

1. Definir el fenómeno o característica a analizar.
2. Seleccionar los factores de estratificación.
3. Clasificar los datos en grupos homogéneos.
4. Representar gráficamente cada grupo homogéneo de datos.
5. Comparar los grupos homogéneos de datos dentro de cada criterio de estratificación.

#### **Usos de la estratificación:**

- 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.

- 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, Feal Cañizares, Coutin Marie, Rodríguez Milord, González Cruz 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.

Un elemento clave para la aplicación de esta metodología 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.

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.

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. También se identificaron los Sistemas de Información Geográfica como herramientas para realizar el proceso de estratificación, que serán analizadas posteriormente.

## 1.2 Técnicas de agrupamiento

Las técnicas de agrupamiento se encargan de descubrir una estructura dentro de un conjunto de datos  $D = \{x_i, \dots, x_n\}$ , dividiéndolo en subconjuntos que muestren ciertas coherencias, es decir, los objetos pueden dividirse en grupos que contienen muestras similares dentro de un mismo grupo ó cluster, 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).

La principal característica de estas técnicas es la utilización de una medida de similaridad 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.

Los métodos de agrupamiento se clasifican en tres grupos fundamentales: jerárquicos, particionales y basados en densidad (Pascual, Pla, Sánchez 2007). 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.

### 1.2.1 Algoritmos jerárquicos

Un algoritmo jerárquico crea una descomposición jerárquica de un conjunto de datos, formando un árbol que divide recursivamente el conjunto de datos, en conjuntos cada vez más pequeños. El árbol puede ser creado de dos formas: de abajo hacia arriba (aglomerativo) o de arriba hacia abajo (divisivo) (Edna 2006).

**CURE** 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 cluster 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, Sánchez 2007).

### **1.2.2 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 objetiva (Edna 2006).

**K-Means** es uno de los algoritmos de agrupamiento más simples, conocidos y eficientes (Pascual, Pla, Sánchez 2007). El mismo 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.

### **1.2.3 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 (Edna 2006).

**DBSCAN** es uno de los primeros algoritmos de agrupamiento que emplea el enfoque de densidad (Pascual, Pla, 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).

### **1.2.4 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).

El algoritmo de agrupamiento basado en la **matriz de similitud** es uno de los más sencillos consultados en la literatura (Francisco José Cortijo Bon 2001). Se basa en la construcción de una matriz de similitud a partir de las distancias entre todas las parejas de patrones.

La matriz de similitud es una matriz cuadrada que se emplea para expresar el grado de similitud 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 similitud 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.

### **1.3 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. 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).

Podrían citarse otras definiciones, pero en esencia pueden concretarse como: sistemas informáticos, con la capacidad de visualizar y analizar información georreferenciada de forma versátil e intuitiva, agilizando la toma de decisiones.

### **1.3.1 Conceptos básicos asociados a los Sistemas de Información Geográfica**

#### **Cartografía**

La cartografía es el arte de trazar mapas geográficos; ciencia que los estudia y se encarga de la elaboración de estos (Lorenzo Martínez 2000). Abarca la creación y el estudio de mapas territoriales y de diferentes dimensiones lineales. También se denomina cartografía a un conjunto de documentos territoriales referidos a un ámbito concreto de estudio.

#### **Datos espaciales**

Los datos espaciales contienen las ubicaciones y formas de características cartográficas. Dentro de su contexto, almacenan información sobre la localización y las formas de un objeto geográfico y las relaciones entre ellos, normalmente con coordenadas y topología.

#### **Mapa**

Un mapa se define como “la representación convencional gráfica de fenómenos concretos o abstractos, localizados en la Tierra o en cualquier parte del Universo” (Ortag 2015).

#### **Mapa topográfico**

Un mapa topográfico es el que representa gráficamente los principales elementos que conforman la superficie terrestre, como vías de comunicación, entidades de población, hidrografía y relieve, con una precisión adecuada a la escala (Ortag 2015).

#### **Mapa temático**

“Un mapa temático es aquel que está diseñado para mostrar características o conceptos particulares” (Ortag 2015).

Un mapa temático se puede definir entonces, como aquel que está diseñado para mostrar fenómenos, eventos, características y conceptos particulares utilizando como base geográfica un mapa topográfico (Torres, Puente 2010).

### **1.3.2 Panorama actual de aplicaciones SIG**

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. A continuación se realiza un breve análisis de algunas de las principales aplicaciones SIG libres existentes. 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.

#### **Software GRASS**

*Geographic Resources Analysis Support System* (GRASS, por sus siglas en inglés) es el proyecto SIG libre más veterano, con un desarrollo de más de 20 años. Su principal característica es su gran número de funcionalidades y su estructura modular favorece que los desarrolladores aporten al proyecto contribuciones individuales centradas en un campo concreto de aplicación.

El mayor problema que presenta es su complejidad y su curva de aprendizaje; aun siendo un software muy potente, carece de una interfaz amigable, y no está diseñado para ser empleado en un entorno de producción. La aparición de herramientas adicionales que facilitan el acceso a la potencia de GRASS, especialmente en el campo del análisis, está cambiando esta situación. Dentro de estas herramientas, Quantum GIS es la más destacable, constituyendo una interfaz de usuario sencilla para GRASS (GRASS Development Team 1998).

#### **Software SAGA**

*System for Automated Geoscientific Analyses* (SAGA, por sus siglas en inglés) es un software SIG de escritorio, multiplataforma, desarrollado en Alemania y con un fuerte enfoque hacia el análisis de datos geoespaciales. SAGA incluye diversos algoritmos y una interfaz de desarrollo que facilita la programación de nuevas funcionalidades de análisis, siendo esta la mayor potencialidad del programa. Otras capacidades, tales como la creación de cartografía o la edición, se encuentran presentes pero muy poco desarrolladas y con escasa funcionalidad, evidenciando que el principal objetivo de este software es servir como herramienta de análisis (J. Böhner 2015).



### Software QGIS

*Quantum GIS* (QGIS, por sus siglas en inglés) es una aplicación SIG con grandes potencialidades para la edición de mapas, multiplataforma y desarrollado utilizando Qt Toolkit<sup>1</sup> y C++. Ofrece muchas características SIG, entre las que se encuentran (QGIS DEVELOPMENT TEAM 2012):

- Permite crear, editar, administrar y exportar mapas vectoriales en varios formatos.
- Permite realizar análisis de datos espaciales de *PostgreSQL/PostGIS*<sup>2</sup> usando el complemento de Python fTools<sup>3</sup>.
- 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<sup>4</sup>.
- Permite la integración de plugins<sup>5</sup> desarrollados en Python a través del módulo PyQGIS.
- Presenta una interfaz amigable.

A partir de las características que presentan las herramientas SIG analizadas, se concluye que 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.

### 1.4 Análisis de soluciones existentes con soporte para la estratificación

En el presente epígrafe se realiza un estudio de las principales herramientas existentes que soportan la realización de procesos de estratificación. Se consideró para el análisis los algoritmos de agrupamiento que estas emplean, la forma de representación de los resultados y los tipos de indicadores que analizan.

---

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.

Los indicadores pueden ser estadísticos o cartográficos en dependencia de la fuente de información que provengan y su tratamiento para el contexto del problema. Los indicadores estadísticos proceden de una fuente de datos estadística. Los indicadores cartográficos describen a la información que se extrae de la cartografía o de otras fuentes donde se especifique la componente espacial asociada. Este tipo de indicadores se relaciona con una localización en el espacio y varía en dependencia de esta.

### **Herramienta para la estratificación de municipios en zonas de riesgo para la salud**

La herramienta es parte del programa *atención integral de los servicios de salud de Hidalgo*; la misma permite realizar estructuraciones de los municipios del estado de Hidalgo de forma automatizada. Brinda al usuario la opción de obtener estratificaciones libres o restringidas a un cierto número de grupos. Se basa en técnicas del Reconocimiento Lógico Combinatorio de Patrones (RLCP), permitiendo manejar diversas formas en la presentación de resultados, en forma gráfica y tabular, así como brindar información acerca de qué indicadores influyen más en la obtención del promedio de riesgo total de cada uno de los grupos formados.

#### Desventajas

- La herramienta es privativa.
- Solo permite estratificar los 84 municipios que conforman el estado Hidalgo.
- Analiza solo indicadores estadísticos.

### **Estratificador INEGI**

La herramienta permite construir agrupaciones o estratificaciones de áreas geográficas en base a información estadística. El sistema brinda al usuario la opción de seleccionar las variables que muestran mayor afinidad con el tema de su interés, y elegir uno o más procedimientos de estratificación; de este modo será posible disponer de dos o más estratificaciones alternativas. Incluye, una serie de ayudas gráficas que permiten al analista realizar comparaciones y decidir cuál de todas las combinaciones de datos y métodos satisface de la mejor manera sus objetivos. Utiliza para la clasificación de los territorios tres algoritmos de agrupamiento, K-medias, Mulvar y MClust, permitiendo representar los resultados mediante diferentes gráficos que pueden ser: mapas temáticos, burbujas, y centroides.

#### Desventajas

- Analiza solo indicadores estadísticos.

- No brinda información sobre el riesgo de salud de los territorios evaluados.
- Tiene definido un conjunto estático de datos, lo que restringe el campo de análisis.
- No permite almacenar las estratificaciones que se realizan.

### SIG para la secretaría de planeación del municipio de Guadalajara de Buga

La herramienta tiene como propósito la organización de la información cartográfica y aprovechar las herramientas que suministran los SIG, para lograr detectar las diversas alteraciones dentro de lo concerniente a usos del suelo, organización vial y riesgos. Este sistema cuenta con un módulo para representar la diferencia social en la distribución de los bienes de acuerdo con los lineamientos del DANE<sup>6</sup>, para ello se aplica una metodología de estratificación logrando la separación de los resultados en estratos y representándolos en mapas temáticos.

#### Desventajas

- Analiza solo indicadores estadísticos.
- Tiene definido un conjunto estático de datos, lo que restringe el campo de análisis.
- No permite almacenar las estratificaciones que se realizan.
- Presenta un conjunto estático de estratos.
- Los datos se agrupan de forma manual.

A continuación se presenta una tabla comparativa entre las soluciones analizadas:

Herramientas Factores de comparación	Herramienta para la estratificación de municipios en zonas de riesgo para la salud	Estratificador INEGI	SIG para la secretaría de planeación del municipio de Guadalajara de Buga
Algoritmos de agrupamiento	MClust, K-Means	K-Means, Mulvar, MClust	Los datos se agrupan de forma manual
Representación gráfica de los resultados	Mapas temáticos	Burbujas, Centroides, Mapas temáticos	Mapas temáticos

6 DANE: Departamento Administrativo Nacional de Estadística de Colombia.

<b>Tipo de indicadores que utiliza</b>	Estadístico	Estadístico	Estadístico
<b>Fuente de información</b>	Obtiene la información a través de una hoja de cálculo	Presenta un conjunto estático de datos	Presenta un conjunto estático de datos
<b>Tipo de licencia</b>	Privativo	Libre	Libre

Tabla 1. Comparación entre los sistemas analizados.

Luego de realizar una comparación entre las herramientas existentes que realizan procesos de estratificación se concluye que presentan deficiencias para incorporar la naturaleza espacial de los datos, y en la obtención de indicadores de diferentes fuentes de datos, además no responden a las particularidades de los centros de salud en Cuba alineadas al paradigma de independencia tecnológica que persigue el país. Estos elementos fundamentan la necesidad de implementar un plugin para el software SIG seleccionado, que incluya la generación de mapas temáticos. Se identificó el algoritmo K-Means como uno de los más utilizados y con mejores resultados en la literatura científica consultada (López Caviedes 2004; Yenisei Bombino Companioni 2005; Erik Limón 2012), por lo que se decide incluir en la propuesta de solución.

## **1.5 Herramientas, lenguajes y tecnologías a utilizar**

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

### **1.5.1 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, Sobernig, Strembeck 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.

### **1.5.2 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 Gonzalez 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, Getting 2010).

Las principales características de la herramienta son:

- Soporta las últimas versiones del UML.
- Posee un poderoso generador de documentación y reportes en formato PDF, HTML y MS Word.
- 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.

### **1.5.3 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 2011).

Se seleccionó Python en su versión 2.7.5 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 librerías disponibles y la potencia del lenguaje, entre otros, hacen que desarrollar una aplicación en Python sea sencillo y rápido.

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.

#### **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.

Utilizando PyQt en su versión 4.0 en el desarrollo de la herramienta informática, se puede crear una interfaz visual sencilla y sin muchos contratiempos, ya que PyQt posee los componentes visuales necesarios para su desarrollo, así como una abundante documentación y ejemplos.

### 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.

### 1.5.4 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<sup>7</sup>, un depurador<sup>8</sup> 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, Mac OS y Linux con una única clave de licencia, también ofrece un espacio de trabajo con colores personalizables y atajos de teclado.

---

<sup>7</sup> Compilador: programa informático que traduce un programa escrito en un lenguaje de programación a otro lenguaje de programación.

<sup>8</sup> Depurador: programa usado para probar y eliminar los errores de otros programas.

La decisión de seleccionar como IDE, Pycharm en su versión 3.4, está dada a que ofrece auto-completación 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).

### 1.5.5 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<sup>9</sup> 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.

---

<sup>9</sup> 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<sup>10</sup> y su integración con el planificador de consultas de PostgreSQL. Utiliza las librerías Proj4<sup>11</sup> para dar soporte a la transformación dinámica de coordenadas y la librería GEOS<sup>12</sup> 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<sup>13</sup> o Unix Domain Sockets (en plataformas Unix), y puede encriptarse mediante SSL<sup>14</sup> para mayor seguridad (Robinson 2011).

### **1.6 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).

---

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

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

12 Geos: librería para trabajar con datos geoespaciales.

13 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.

14 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).

### **1.6.1 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 realimentación 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. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (Joskowicz 2008).

#### **Características de la metodología XP (Beck 2000):**

- XP es una metodología “liviana” 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**

Con el desarrollo de este capítulo se obtuvo un mejor dimensionamiento del problema a partir del análisis de los principales conceptos asociados a la solución. El estudio de las técnicas de agrupamiento facilitó

identificar las más utilizadas y eficientes, decidiendo incluir a la solución el algoritmo de agrupamiento basado en la matriz de similitud. La revisión del panorama actual de los softwares SIG condujo a la selección de QGIS como el más adecuado para la integración de la solución. El estudio de varios sistemas existentes que soportan procesos de estratificación facilitó obtener una mejor visión de la solución, aportó nuevas funcionalidades que podrían ser de interés y permitió identificar el algoritmo K-Means como uno de los más utilizados y con mejores resultados, por lo que se decide incluir también en la propuesta. Para la implementación del software fue seleccionado un conjunto de herramientas y tecnologías basadas en licencias de software libre, encaminado a obtener un producto de alta independencia tecnológica y utilizable en diferentes plataformas. Finalmente se escogió la metodología XP para guiar el proceso de desarrollo de la solución.

## CAPÍTULO 2: Propuesta de solución para la estratificación de territorios

En este capítulo se presenta una propuesta de solución para realizar el proceso de estratificación de territorios y se describe cada una de las fases que la conforman. 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 del proceso de estratificación

En el presente trabajo el proceso de estratificación se desglosa en las fases siguientes:

- Selección: se eligen los factores de estratificación (territorios, indicadores).
- Pre-procesamiento: se obtiene el aporte informacional y se normalizan los datos de los indicadores seleccionados.
- Agrupamiento: se clasifican los territorios en grupos homogéneos.
- Visualización: se representa en un mapa temático cada grupo homogéneo de territorios.

En la [figura 1](#) se presenta el modelo conceptual de la propuesta de solución.

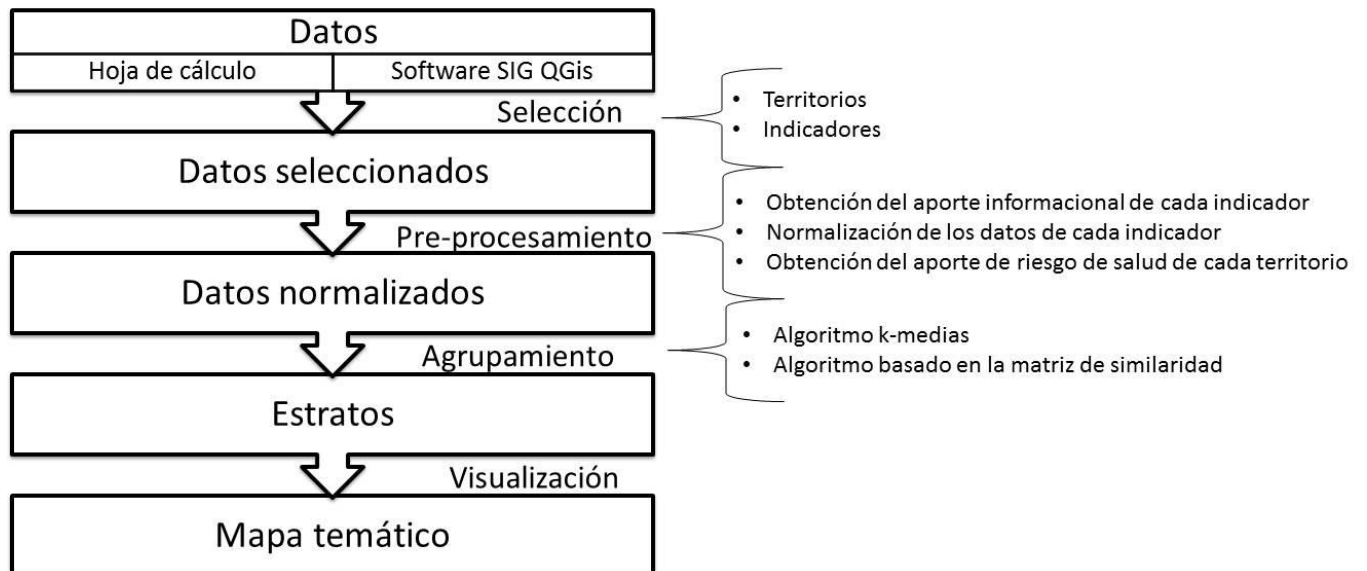


Figura 1. Modelo conceptual de la propuesta de solución.

Para realizar el proceso de estratificación se propone la utilización de indicadores estadísticos definidos 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.

Se tuvo en consideración:

- el problema de contaminación de las aguas, este se encuentra entre los principales problemas ambientales del país. Ello se debe principalmente al impacto producido en áreas densamente pobladas y las alteraciones a la salud y a la calidad de vida de la población (CubaEduca 2015).
- el problema de la contaminación del aire, este 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 (Cuba enfrenta la contaminación ambiental | Isla al Sur 2015).
- 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 los distintos indicadores estudiados en la literatura (Yenisei Bombino Companioni 2005) están asociados a este tipo de datos.

### **2.1.1 Pre-procesamiento de los datos**

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

La aportación informacional de los indicadores se obtiene mediante un método estadístico 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).

#### Varianza de los datos

$$S^2 = \frac{1}{t} \sum_{i=1}^t (V_i - \bar{x})^2$$

Donde:

- $t$ : Total de la población finita de los datos.

- $\bar{x}$ : Promedio de los datos.
- $V_i$ : Valor  $i$ , donde  $t= 1,2,\dots,t$ .

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.

El coeficiente de variación se determina de la siguiente manera:

$$W_k = \frac{\text{desviación estándar}}{\text{media}} = \frac{S}{\bar{x}}$$

Por lo tanto,  $W_k$  es la aportación informativa del indicador  $k$ , donde  $k=1,2,\dots$ , total de indicadores.

### Normalización de los datos seleccionados

Para evitar que un indicador no domine sobre otro, sus valores se normalizaron, con la siguiente función (Molina López, García Herrero 2006).

$$\frac{X_{if} - \text{Min}_f}{\text{Max}_f - \text{Min}_f}$$

Donde:

- $X_{if}$ : Valor  $i$  del indicador  $f$ , donde  $i=1,2,\dots$ , total de territorios.
- $\text{Min}_f$ : Mínimo valor del indicador  $f$ .
- $\text{Max}_f$ : Máximo valor del indicador  $f$ .
- $f=1,2,\dots$ , total de indicadores.

### Medida de similitud empleada

Para determinar la semejanza entre los territorios se utiliza la métrica *distancia euclidiana ponderada*. Esta medida de similitud se identifica como una de las más utilizadas y sencillas (Edna 2006). En esta métrica cuando los valores son numéricos, se obtienen resultados satisfactorios en la clasificación (Rodríguez, Blanco, Camacho 2013).

**Distancia euclidiana ponderada** (Berzal 2004):

$$d(X_i, X_j) = \sqrt{\sum_{k=1}^p W_k (X_{ik} - X_{jk})^2}$$

Donde:

- $X_i$ : Territorio  $i$
- $X_j$ : Territorio  $j$ .
- $W_k$  : Es el aporte informacional del indicador  $k$  .
- $p$ : Total de indicadores.
- $X_{ik}$ : Valor de  $X_i$  en el indicador  $k$ .
- $X_{jk}$ : Valor de  $X_j$  en el indicador  $k$ .
- $k = 1, 2, \dots, p$ .
- $i, j = 1, 2, \dots$ , total de territorios.

### 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.

**Caso#1: A mayor valor en el dato implica mayor aportación de riesgo para el indicador** (Alba-Cabrera 1997).

$$Ap = \frac{X_{ik}}{Max_k}$$

Donde:

- $Ap$ : Aportación de riesgo para el caso 1.
- $X_{ik}$ : Es el valor que tiene el territorio  $X_i$ ,  $i=1, \dots, h$ , donde  $h$  es el total de territorios, en el indicador  $k$ ,  $k= 1, \dots, n$ , donde  $n$  es el total de indicadores.
- $Max_k$ : Es el mayor valor que toma el indicador  $k$ .



**Caso#2: A mayor valor en el dato implica menor aportación de riesgo para el indicador** (Alba-Cabrera 1997).

$$Ap = 1 - \frac{X_{ik}}{Max_k}$$

Donde:

- $Ap$ : Aportación de riesgo para el caso 2.
- $X_{ik}$ : Es el valor que tiene el territorio  $X_i$ ,  $i=1, \dots, h$ , donde  $h$  es el total de territorios, en el indicador  $k$ ,  $k= 1, \dots, n$ , donde  $n$  es el total de indicadores.
- $Max_k$ : Es el mayor valor que toma el indicador  $k$ .

Finalmente se aplica una función de integración de los indicadores para obtener la aportación de riesgo final por territorio.

**Función total de aportaciones de riesgo** (Alba-Cabrera 1997).

$$\Gamma_{TO}(X_m) = \frac{\sum_{i=1}^{nm} W_i Ap_i}{\sum_{i=1}^{nm} W_i}$$

Donde:

- $\Gamma_{TO}$ : Es la aportación de riesgo por territorio.
- $X_m$ : Territorio  $m$ , donde  $m = 1, 2, \dots$ , total de territorios.
- $Ap_i$ : Aportación de riesgo del indicador  $i$ .
- $W_i$ : Aportación informativa del indicador  $i$ .
- $nm$ : Total de indicadores.
- $i = 1, 2, \dots, nm$ .

### 2.1.2 Agrupamiento de los datos

Para clasificar los territorios en grupos homogéneos (estratos), se proponen utilizar los algoritmos de agrupamiento de datos que se describen a continuación.

---

### Algoritmo k-medias

---

#### Entradas:

- Un conjunto de N patrones  $\{ X_1, \dots, X_n \}$
- Número K de agrupaciones.

#### Salidas:

- Los  $S_1, S_2, \dots, S_k$  agrupamientos formados
- Los centros  $Z_1, Z_2, \dots, Z_k$  de los agrupamientos.

#### Pasos

1. Escoger los K centroides aleatoriamente
  2. Inicializar los K centros  $Z_i(t)$ ,  $i=1,2,\dots, k$
  3. Formar K grupos, asignando cada punto al centroide más cercano
  4.  $t \leftarrow t+1$
  5. Para cada  $X = X_1, X_2, \dots, X_n$
  6. Asignar cada X al grupo más cercano
  7.  $m \leftarrow \text{GrupoMasCercano}(X)$
  8.  $S_m(t) \leftarrow S_m(t-1) \cup \{ X \}$
  9. Fin Para
  10. Para  $i=1,2,\dots,k$
  11. Recalcular los centros
  12.  $Z_i(t) \leftarrow \text{RecalcularCentro}(i)$
  13. Fin Para
  14. Si  $Z_i(t) = Z_i(t-1)$ , para  $i=1,2,\dots,k$
  15. Terminar
  16. Si-no Algún  $Z_i(t) \neq Z_i(t-1)$
  17. Ir al paso 3
  - 18:Fin-Si
-

---

### **Algoritmo Matriz de Similaridad**

---

#### **Entradas:**

- Un conjunto de M patrones  $\{X_1, \dots, X_m\}$
- Umbral de distancias  $\Theta$

#### **Salidas:**

- Número de agrupamientos A encontrados,
- Conjuntos de agrupamientos  $S_1, S_2, \dots, S_A$
- Centros los agrupamientos  $Z_1, Z_2, \dots, Z_A$ .

#### **Auxiliares:**

- Matriz de similaridad S.

#### **Pasos**

1. Construir la matriz de similaridad S
  2. Inicializar  $A=0$
  3. Seleccionar fila de S que contenga más unos. (Si hay más de una, seleccionarla arbitrariamente)
  4. Sea i la fila seleccionada
  5. Crear agrupamiento formado por  $X_i$  y todos los patrones  $X_j$  tales  $S_{ij}=1$
  6.  $A \leftarrow A+1$
  7. Para todo  $X_j$  del agrupamiento considerado
  8. Si  $S_{jk}=1$
  9. Añadir el patrón  $X_k$  al agrupamiento
  10. Fin-Si
  11. Fin-Para
  12. Repetir desde el paso 7 al 10, hasta que no se pueda añadir nuevos patrones indirectos al agrupamiento.
  13. Eliminar todas las filas y columnas de S correspondientes a los patrones que se han incorporados al agrupamiento i para formar una nueva matriz S reducida
  14. Repetir desde el paso 7 al 13, hasta que no sean posibles más reducciones
-

## **2.2 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 repercutiendo en el resto de las fases de desarrollo del mismo. Además contribuye a tomar mejores decisiones de diseño y de arquitectura.

### **2.2.1 Requisitos 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).

A continuación se muestran los RF identificados:

- **RF 1:** Importar indicadores estadísticos desde una hoja de cálculo.
- **RF 2:** Obtener características cartográficas a través de QGIS.
- **RF 3:** Construir estratificación.
  - **RF 3.1:** Construir estratos.
  - **RF 3.2:** Visualizar estratos construidos en mapa temático
- **RF 4:** Gestionar las estratificaciones.
  - **RF 4.1:** Adicionar estratificación.
  - **RF 4.2:** Mostrar estratificación.
  - **RF 4.3:** Eliminar estratificación.
- **RF 5:** Exportar mapa temático de una estratificación como imagen.
- **RF 6:** Exportar estratificación hacia una hoja de cálculo.

### 2.2.2 Requisitos no funcionales

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 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.

### 2.3 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. También el cliente

especifica la prioridad en que se deben implementar las historias de usuario, además de una estimación del esfuerzo que costará. 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 el desarrollo del mismo (Beck 2000).

### 2.3.1 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 nueve HU. En las tablas [2](#) y [3](#) se muestra una breve descripción de dos de ellas, el resto se encuentran especificadas en [anexos](#).

Historia de Usuario “Construir estratos”	
<b>Número:</b> 3.1	<b>Nombre Historia de Usuario:</b> Construir estratos
<b>Usuario:</b> Experto	
<b>Prioridad en Negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Alto
<b>Puntos Estimados:</b> 21	<b>Iteración Asignada:</b> 1
<b>Programador responsable:</b> Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> La aplicación debe ser capaz de construir estratos, a partir de: <ul style="list-style-type: none"><li>• los indicadores estadísticos importados por el usuario desde una hoja de cálculo.</li><li>• las características cartográficas seleccionadas por el usuario a través de QGIS.</li></ul>	
<b>Observaciones:</b>	

Tabla 2. Historia de usuario: Construir estratos.

Historia de Usuario “Exportar estratificación hacia una hoja de cálculo”	
<b>Número:</b> 6	<b>Nombre Historia de Usuario:</b> Exportar estratificación hacia una hoja de cálculo.
<b>Usuario:</b> Experto	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo en Desarrollo:</b> Media
<b>Puntos Estimados:</b> 3	<b>Iteración Asignada:</b> 4
<b>Programador responsable:</b> Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> La aplicación debe ser capaz de exportar los resultados de una estratificación construida hacia una hoja de cálculo.	
<b>Observaciones:</b>	

Tabla 3. Historia de usuario: Exportar estratificación hacia una hoja de cálculo.

### 2.3.2 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.

Historia de usuario	Puntos de estimación (semanas)
<b>HU 1:</b> Importar indicadores estadísticos desde una hoja de cálculo	1
<b>HU 2:</b> Obtener características cartográficas a través de QGis.	1
<b>HU 3.1:</b> Construir estratos.	3
<b>HU 3.2:</b> Visualizar estratos construidos en un mapa temático.	3
<b>HU 4.1:</b> Adicionar estratificación	1
<b>HU 4.2:</b> Mostrar estratificación	1
<b>HU 4.3:</b> Eliminar estratificación	1
<b>HU 5:</b> Exportar mapa temático de una estratificación como imagen.	0.5
<b>HU 6:</b> Exportar estratificación hacia una hoja de cálculo.	0.5

Tabla 4. Estimación de esfuerzos por Historia de Usuario.

### 2.3.3 Plan de iteraciones

Una vez finalizadas las HU se debe crear un plan de iteraciones, indicando cuáles se desarrollarán en cada iteración del programa. En la tabla 5 se muestra cómo quedó definido el plan de iteraciones de la solución propuesta.

Iteraciones	Orden de las historias de usuario a implementar	Duración de las iteraciones (semanas)
Iteración 1	Construir estratos	3
Iteración 2	Visualizar estratos contruidos en mapa temático.	3
Iteración 3	Gestionar estratificaciones.	3
Iteración 4	Importar indicadores estadísticos desde una hoja de cálculo.	2
	Obtener características cartográficas a través de QGIS.	
Iteración 5	Exportar mapa temático de una estratificación como imagen.	1
	Exportar estratificación hacia una hoja de cálculo.	
Total		12

Tabla 5. Plan de duración de las iteraciones.

### 2.3.4 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.

	Final de la 1ra Iteración	Final de la 2da Iteración	Final de la 3ra Iteración	Final de la 4ta Iteración	Final de la 5ta Iteración
<b>Módulos</b>	1ra semana de marzo	4ta semana de marzo	3ra semana de abril	1ra semana de mayo	2da semana de mayo
<b>Estratificación</b>	v1.0	v1.1	v1.2	v1.3	Finalizado

Tabla 6. Plan de duración de las entregas.



## 2.4 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).

### 2.4.1 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, Reinaga 2008).

En las tablas [7](#) y [8](#) se muestran las tarjetas CRC correspondientes a las clases *Territorio* y *Estratificación*.

Clase: Territorio	
Responsabilidad	Colaboración
<ul style="list-style-type: none"><li>• Calcular el aporte de riesgo de cada territorio.</li><li>• Crear instancias de la clase Indicador.</li></ul>	Estrato, Indicador, Algoritmo

Tabla 7. Tarjeta CRC para la clase Territorio.

Clase: Estratificación	
Responsabilidad	Colaboración
<ul style="list-style-type: none"><li>• Crear instancias de la clase Estrato.</li></ul>	ControladorEstratificador, Estrato

Tabla 8. Tarjeta CRC para la clase Estratificación.

## 2.5 Arquitectura de software

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).

### 2.5.1 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). En este trabajo se hace uso del estilo arquitectónico en capas, logrando que el sistema quede organizado y así tener un orden lógico en la programación del mismo.

#### 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 2](#) se presenta una imagen de la arquitectura de la solución.

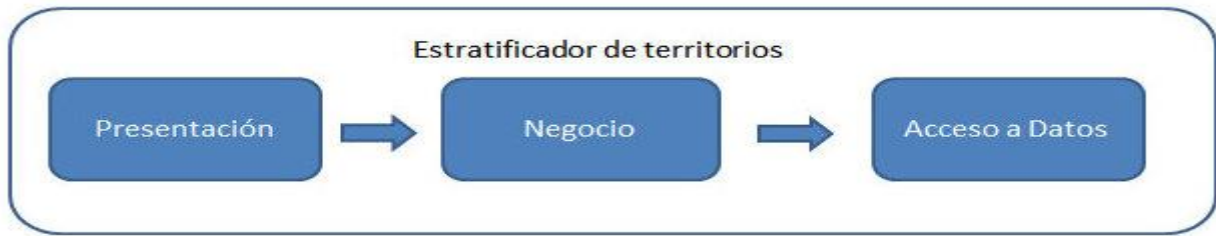


Figura 2. Evidencia de la arquitectura en capas.

**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.

**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.

**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.

## 2.6 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.

### 2.6.1 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 3](#) se muestra una imagen de dicha clase.

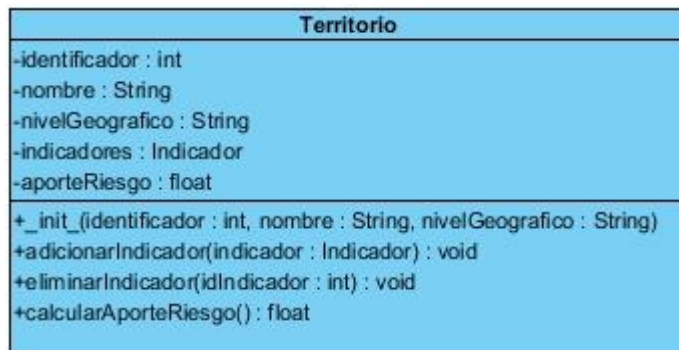


Figura 3. Evidencia del patrón Experto.

**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 4](#) se presenta la evidencia de dicho patrón.

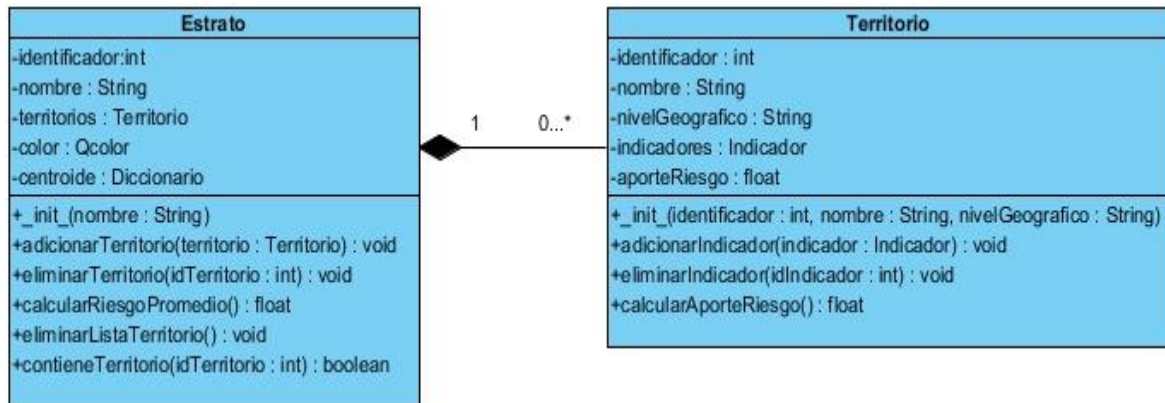


Figura 4. Evidencia del patrón Creador.

**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.

**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 en conjunto con el patrón bajo acoplamiento, de forma tal que cada clase realice sus acciones y se evita que otra clase realice acciones correspondientes a la clase con la que está relacionada.

**Controlador:** Permite asignar la responsabilidad de controlar el flujo de *eventos del sistema*<sup>15</sup>, 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

<sup>15</sup> Evento del sistema: Es un evento de alto nivel generado por un actor externo. Es un evento de entrada externa.

se le asignó la responsabilidad de manejar los eventos del sistema generados por el usuario. En la [figura 5](#) se muestra una imagen de dicha clase.



Figura 5. Evidencia del patrón Controlador.

### 2.6.2 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 Rodriguez 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 Rodriguez 2010).
- **Creacionales:** Permiten abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados (Eric Rodriguez 2010).
- **Estructurales:** Describen cómo las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades (Eric Rodriguez 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 6](#) se muestra cómo se evidencia el patrón plantilla en la aplicación informática propuesta.

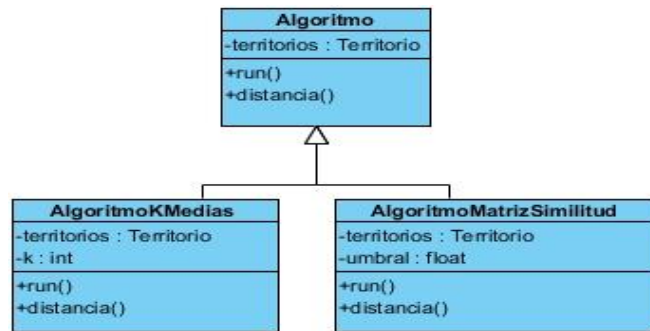


Figura 6. Evidencia del patrón Plantilla.

## 2.7 Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación (Larman 1999). Un diagrama de este tipo presenta las clases del sistema con sus relaciones estructurales y de herencia. En la [figura 7](#) se muestra el diagrama de clases de la aplicación informática propuesta.

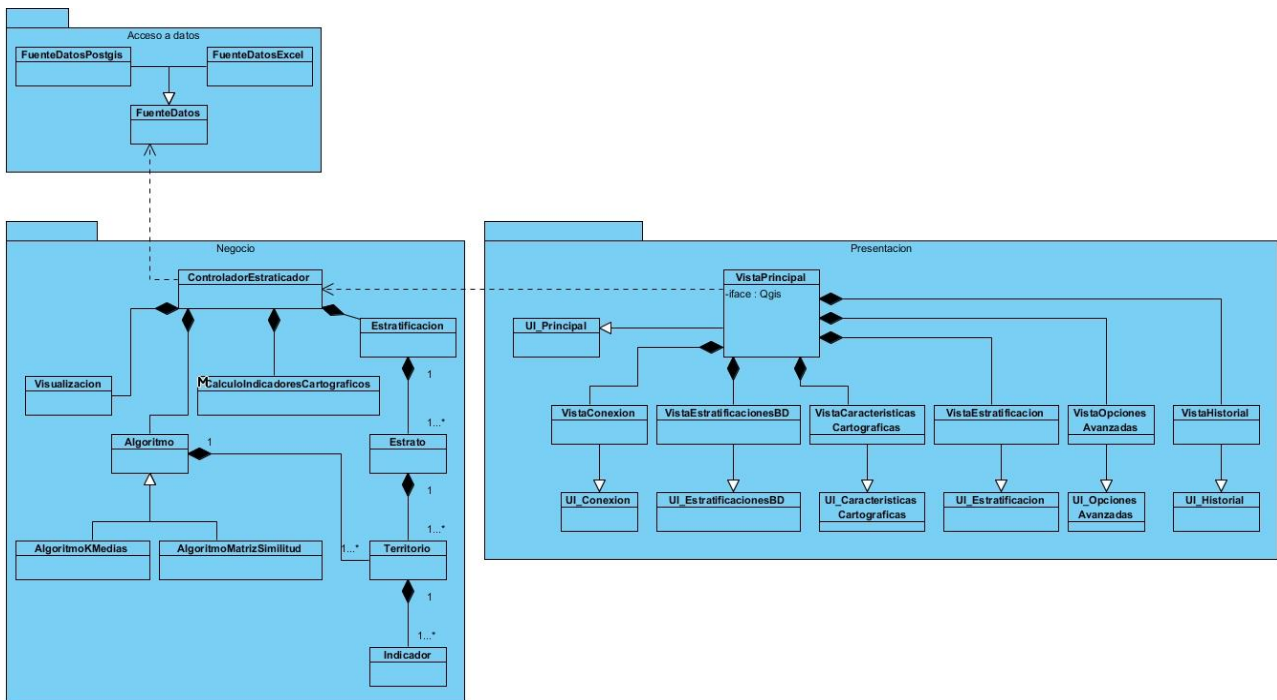


Figura 7. Diagrama de clases.



## 2.8 Diagrama Entidad-Relación

Los Diagramas Entidad-Relación (DER) definen todos los datos que se introducen, se almacenan, se transforman y se producen dentro de una aplicación (Pressman, 2006). Estos modelos representan a la realidad a través de un esquema gráfico empleando la terminología de Entidades, que son objetos que existen y son los elementos principales que se identifican en el problema a resolver. En la [figura 8](#) se muestra el diagrama Entidad-Relación de la aplicación informática propuesta.

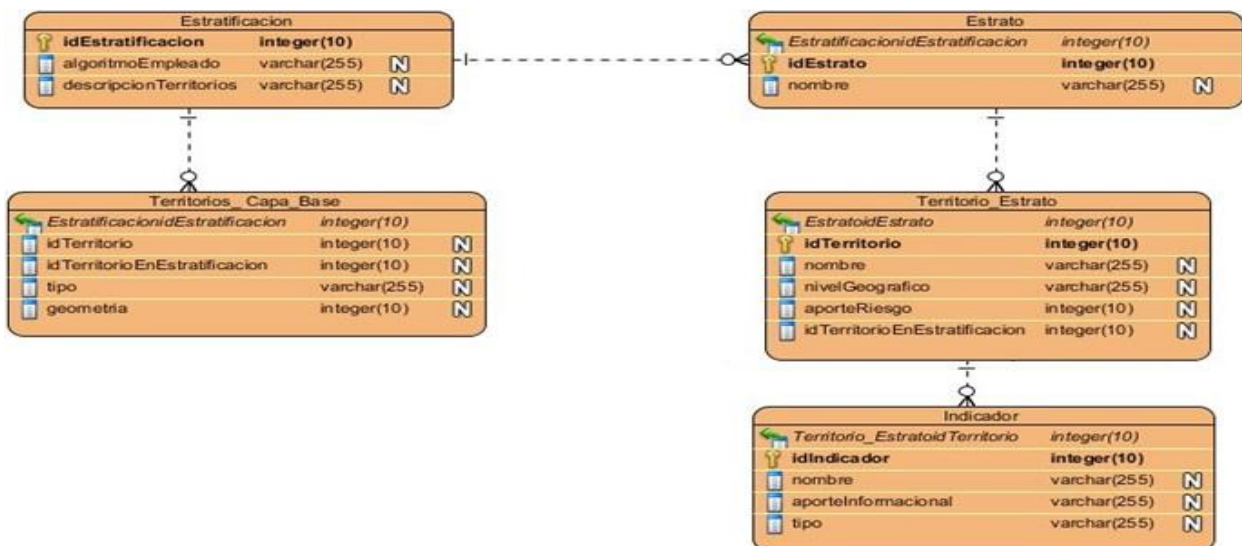


Figura 8. Diagrama Entidad-Relación.

## Conclusiones parciales

La propuesta de solución definida facilitó realizar el proceso de estratificación de territorios utilizando SIG. La identificación de los requisitos permitió un mayor entendimiento de las necesidades del cliente. Mediante la descripción de las HU divididas por iteraciones y 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 y de los patrones de diseño GRASP Y GoF permitió una mejor estructuración de la aplicación.



## CAPÍTULO 3: Implementación y validación de la solución

En el siguiente capítulo se detallan las tareas de ingenierías que conforman cada HU definida en la fase de planificación. Se define el estándar de codificación que se estará utilizando en el desarrollo de la solución. Además se realizan las pruebas definidas por la metodología seleccionada, así como 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. Se realiza un caso de estudio para verificar la validez de los resultados de la solución propuesta.

### 3.1 Fase de implementación

Luego de haber definido los elementos necesarios en la etapa de planificación y diseño se pasa a la de codificación o implementación de la aplicación, donde se da cumplimiento al plan de iteraciones. En esta fase se realiza la implementación de las HU que fueron seleccionadas por cada iteración, además se crean las tareas de ingeniería para ayudar a organizar la implementación exitosa de las HU. Al finalizar esta fase el cliente estará listo para comenzar a 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 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](#).

HU	Tareas de ingeniería por HU
Construir estratos	<ul style="list-style-type: none"><li>• Agrupar los territorios utilizando el algoritmo k-medias.</li><li>• Agrupar los territorios utilizando el algoritmo basado en la matriz de similaridad.</li><li>• Obtener el aporte informacional de los indicadores.</li><li>• Obtener el aporte de riesgo de salud de los territorios.</li><li>• Normalizar los datos de los indicadores.</li></ul>

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

Tarea de ingeniería	
Número Tarea: 1	Número Historia de Usuario: HU # 3.1
Nombre Tarea: Agrupar territorios utilizando el algoritmo k-medias	
Tipo Tarea: Desarrollo	Puntos Estimados: 1
Fecha Inicio: 16/02/2015	Fecha Fin: 20/02/2015
Programador Responsable: Rolando Morales Pérez, Yanislay Torres Vega	
Descripción: Esta tarea permite agrupar los territorios seleccionados por el usuario utilizando el algoritmo k-medias.	

Tabla 10. Tarea de Ingeniería Agrupar territorios utilizando el algoritmo k-medias.

### 3.1.1 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 se 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**

Cuando se desarrolla una solución informática se deben realizar una gran cantidad de pruebas para verificar que el código esté correcto. Estas pruebas normalmente 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 características cartográficas a través de QGIS* y *Construir estrato*.

Caso de prueba de aceptación	
<b>Código:</b> HU2_P1	<b>Historia de Usuario:</b> 2
<b>Nombre:</b> Obtener características cartográficas a través de QGIS.	
<b>Descripción:</b> Prueba para validar la funcionalidad obtener características cartográficas a través de QGIS.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"> <li>• El usuario debe escoger la opción <i>Seleccionar características cartográficas</i>.</li> <li>• El usuario debe seleccionar una capa <i>para la identificación de los territorios a evaluar (capaT)</i>.</li> <li>• El usuario debe seleccionar un atributo de la <u>capaT</u> para nombrar los territorios que se desean evaluar.</li> <li>• El usuario puede seleccionar una capa para la identificación de los ríos que presentan contaminación.</li> <li>• El usuario si selecciona una capa para la identificación de los ríos que presentan contaminación debe seleccionar un atributo de esta para nombrar sus características.</li> <li>• El usuario si selecciona una capa para la identificación de los ríos contaminados debe seleccionar al menos un río que presente contaminación.</li> <li>• El usuario puede seleccionar una capa para la identificación de los puntos contaminantes.</li> <li>• El usuario debe seleccionar la opción <i>Aceptar</i></li> </ul>	
<b>Resultados esperados:</b> En caso que se cumplan las condiciones de ejecución, el sistema obtiene las características cartográficas seleccionadas. En caso contrario el sistema muestra un mensaje informando el motivo por el cual no obtuvo las características cartográficas seleccionadas por el usuario.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria	

Tabla 11. Caso de prueba de aceptación Obtener características cartográficas a través de QGIS.

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU3.1_P1	<b>Historia de Usuario:</b> 3.1
<b>Nombre:</b> Construir estrato.	
<b>Descripción:</b> Prueba para validar la funcionalidad construir estrato.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"> <li>• El sistema debe haber obtenido la capa para los territorios a evaluar seleccionada por el usuario.</li> <li>• El sistema debe haber obtenido los datos de los indicadores estadísticos y/o los datos de los indicadores cartográficos.</li> <li>• El usuario debe seleccionar la opción <i>Estratificar</i>.</li> <li>• El usuario debe seleccionar un algoritmo de agrupamiento.</li> <li>• El usuario debe seleccionar al menos un territorio a evaluar.</li> <li>• El usuario debe seleccionar el nivel geográfico de los territorios a evaluar.</li> <li>• Si el usuario selecciona el algoritmo de agrupamiento K-Medias, la cantidad de estratos resultantes debe ser menor o igual que la cantidad de territorios seleccionados (la cantidad de estratos resultantes se selecciona en <i>Opciones Avanzadas</i>).</li> <li>• El usuario debe seleccionar al menos un indicador.</li> <li>• El usuario debe seleccionar un criterio de riesgo de salud para cada indicador elegido.</li> <li>• El usuario debe seleccionar la opción <i>Aceptar</i>.</li> </ul>	
<b>Resultados esperados:</b> En caso que se cumplan las condiciones de ejecución, el sistema construye los estratos. En caso contrario el sistema muestra un mensaje informando el motivo por el cual no realizó la estratificación.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria	

Tabla 12. Caso de prueba de aceptación Construir estrato.

### **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 9 casos de prueba de aceptación en conjunto cliente-desarrolladores. De este total, 6 arrojaron el resultado esperado mientras que 3 pruebas resultaron fallidas, las funcionalidades que respondían a estas pruebas fueron tratadas en la siguiente iteración y al volver a aplicar las pruebas de

funcionalidad mostraron un resultado exitoso. Finalmente se obtuvieron un total 9 pruebas satisfactorias de 9 casos de prueba aplicados.

### 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 internas.

La técnica utilizada dentro de las pruebas de caja blanca fue camino básico. En la [figura 9](#) se enumeran las sentencias de código del método `runCR()`.

```
def runCR(self):
    diccionario={}
    for territorio in self.territoriosSeleccionados:
        poligonoFeature=self.getFeatureI(territorio)
        cantidad=0
        for rio in self.linesSeleccionados:
            lineFeature=self.getFeatureR(rio)
            t=[]
            t=poligonoFeature.geometry().intersection(lineFeature.geometry())
            if len(t.asPolyline())>0:
                cantidad+=1
        diccionario[territorio]=cantidad
    self.controladora.matrizValoreCR=diccionario
```

Figura 9. Código del método `runCR()`.

Luego de haberse construido el grafo se realiza el cálculo de la *complejidad ciclomática*<sup>16</sup> 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  $V(G) = \text{Aristas} - \text{Nodos} + 2$ .
3. La complejidad ciclomática,  $V(G)$ , de un grafo de flujo  $G$ , también se define como  $V(G) = \text{Nodos de predicado}^{17} + 1$ .

A partir del grafo de flujo del método  $\text{runCR}()$  que se presenta en la [figura 10](#), la complejidad ciclomática sería:

- Como el grafo tiene cuatro regiones,  $V(G) = 4$
- Como el grafo tiene 11 aristas y 9 nodos,  $V(G) = 11 - 9 + 2 = 4$
- Como el grafo tiene 3 nodos de predicado,  $V(G) = 3 + 1 = 4$

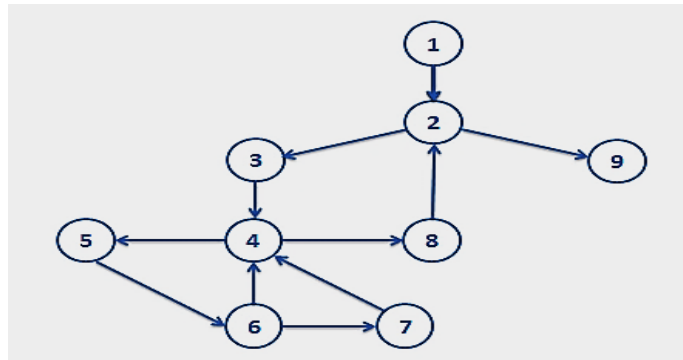


Figura 10. Grafo de flujo del método  $\text{runCR}()$ .

Dado a que el cálculo de las tres fórmulas anteriormente mencionadas arrojó el mismo resultado se puede plantear que la complejidad ciclomática del método es 4. Esto significa que existen 4 posibles caminos por donde el flujo puede circular. Este valor representa el número mínimo de casos de pruebas para el procedimiento tratado.

<sup>16</sup> Complejidad ciclomática: es una métrica del software que proporciona una medición cuantitativa de la complejidad lógica de un programa.

<sup>17</sup> Nodo predicado: es él que representa una condicional if o case, es decir, de él salen varios caminos.

Caminos básicos identificados:

- Camino 1: 1-2-9
- Camino 2: 1-2-3-4-8-2-9
- Camino 3: 1-2-3-4-5-6-4-8-2-9
- Camino 4: 1-2-3-4-5-6-7-4-8-2-9

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

<b>Caso de Prueba para el camino básico #1 (1-2-9)</b>	
<b>Descripción</b>	Prueba para comprobar los resultados de la función runCR() en caso que la lista de territorios a evaluar sea vacía.
<b>Condición de ejecución</b>	<ul style="list-style-type: none"><li>• longitud de self.territoriosSeleccionados = 0</li></ul>
<b>Entrada</b>	<ul style="list-style-type: none"><li>• self.territoriosSeleccionados=[ ]</li><li>• self.linesSeleccionados=[ ]</li></ul>
<b>Resultado</b>	self.controladora.matrizValoreCR = { }
<b>Resultado de la prueba</b>	Prueba satisfactoria

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

<b>Caso de Prueba para el camino básico #2 (1-2-3-4-8-2-9)</b>	
<b>Descripción</b>	Prueba para validar los resultados de la función runCR() en caso que la lista de ríos contaminados sea vacía.
<b>Condición de ejecución</b>	<ul style="list-style-type: none"><li>• longitud de self.territoriosSeleccionados &gt; 0</li><li>• longitud de self.linesSeleccionados = 0</li></ul>
<b>Entrada</b>	<ul style="list-style-type: none"><li>• self.linesSeleccionados=[ ]</li><li>• self.territoriosSeleccionados=[2,3]</li></ul>
<b>Resultado</b>	self.controladora.matrizValoreCR = { }
<b>Resultado de la prueba</b>	Prueba satisfactoria

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



<b>Caso de Prueba para el camino básico #3 (1-2-3-4-5-6-4-8-2-9)</b>	
<b>Descripción</b>	Prueba para validar los resultados de la función runCR() en caso que no existan intersecciones entre los ríos y los territorios a evaluar.
<b>Condición de ejecución</b>	<ul style="list-style-type: none"> <li>• longitud de self.territoriosSeleccionados &gt; 0</li> <li>• longitud de self.linesSeleccionados &gt; 0</li> <li>• longitud de t.asPolyline() = 0</li> </ul>
<b>Entrada</b>	<ul style="list-style-type: none"> <li>• self.linesSeleccionados=[1,2]</li> <li>• self.territoriosSeleccionados=[2,3]</li> </ul>
<b>Resultado</b>	self.controladora.matrizValoreCR = { 2: 0, 3: 0}
<b>Resultado de la prueba</b>	Prueba satisfactoria

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

<b>Caso de Prueba para el camino básico #4 (1-2-3-4-5-6-7-4-8-2-9)</b>	
<b>Descripción</b>	Prueba para validar los resultados de la función runCR() en caso que existan intersecciones entre los ríos y los territorios a evaluar.
<b>Condición de ejecución</b>	<ul style="list-style-type: none"> <li>• longitud de self.territoriosSeleccionados &gt; 0</li> <li>• longitud de self.linesSeleccionados &gt; 0</li> <li>• longitud de t.asPolyline() &gt; 0</li> </ul>
<b>Entrada</b>	<ul style="list-style-type: none"> <li>• self.linesSeleccionados=[1,2]</li> <li>• self.territoriosSeleccionados=[2,3]</li> </ul>
<b>Resultado</b>	self.controladora.matrizValoreCR = { 2: 5, 3: 4}
<b>Resultado de la prueba</b>	Prueba satisfactoria

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

### **Análisis de los resultados**

Para la validación del código generado en el desarrollo de la herramienta se seleccionaron los métodos más relevantes, a los cuales se le realizaron las pruebas para poder evaluar si el funcionamiento de cada uno de estos métodos se comportó de la manera esperada. Se realizaron un total de 11 pruebas a las 5 funcionalidades seleccionadas como relevantes, de las cuales 8 resultaron satisfactorias en una primera iteración de pruebas y 3 no satisfactorias. Estas últimas fueron solucionadas en una segunda iteración

para obtener un 100% de pruebas satisfactorias, comprobándose la estabilidad de la lógica aplicada en el código generado en el desarrollo de la herramienta informática.

### **3.3 Caso de estudio**

Con el fin de valorar los resultados de la solución propuesta se decide aplicar un caso de estudio, en correspondencia con el trabajo realizado por (Yenisei Bombino Companioni 2005), donde se realiza un proceso de estratificación de las catorce provincias de Cuba definidas en la división política-administrativa de 1976. Se utilizó como fuente de información el Anuario Estadístico del año 2001 (INFOMED 2001) y se seleccionaron los indicadores siguientes:

- Mortalidad infantil por cada 1000 nacidos vivos.
- Mortalidad infantil de los niños menores de cinco años por cada 1000 nacidos vivos.
- Mortalidad por enfermedades del corazón por cada 100 000 habitantes.
- Mortalidad por tumores malignos por cada 100 000 habitantes.
- Mortalidad por enfermedades cerebrovasculares por cada 100 000 habitantes.
- Mortalidad por influenza y neumonía por cada 100 000 habitantes.
- Mortalidad por accidentes por cada 100 000 habitantes.
- Mortalidad perinatal por cada 1000 nacidos vivos.
- Mortalidad por enfermedades infecciosas y parasitarias por cada 100 000 habitantes.
- Mortalidad materna por cada 100 000 nacidos vivos.
- Incidencia de tuberculosis por cada 100 000 habitantes
- Incidencia de hepatitis por cada 100 000 habitantes.
- Incidencia de diabetes por cada 100 000 habitantes.
- Incidencia de hipertensión por cada 100 000.
- Incidencia de asma por cada 100 000 habitantes.
- Incidencia de bajo peso al nacer.
- Consultas médicas por habitante.
- Ingresos por cada 100 habitantes.
- Camas de asistencia por cada 1000 habitantes.

- Consultas de puericultura por habitante.
- Consultas de pediatría por habitante.
- Densidad poblacional.
- Población mayor de 60 años.
- Población menor de 1 año.
- Población menor de 15 años.
- Natalidad por cada 1000 habitantes.

### Aplicación del 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 11](#) muestra parte del proceso realizado.

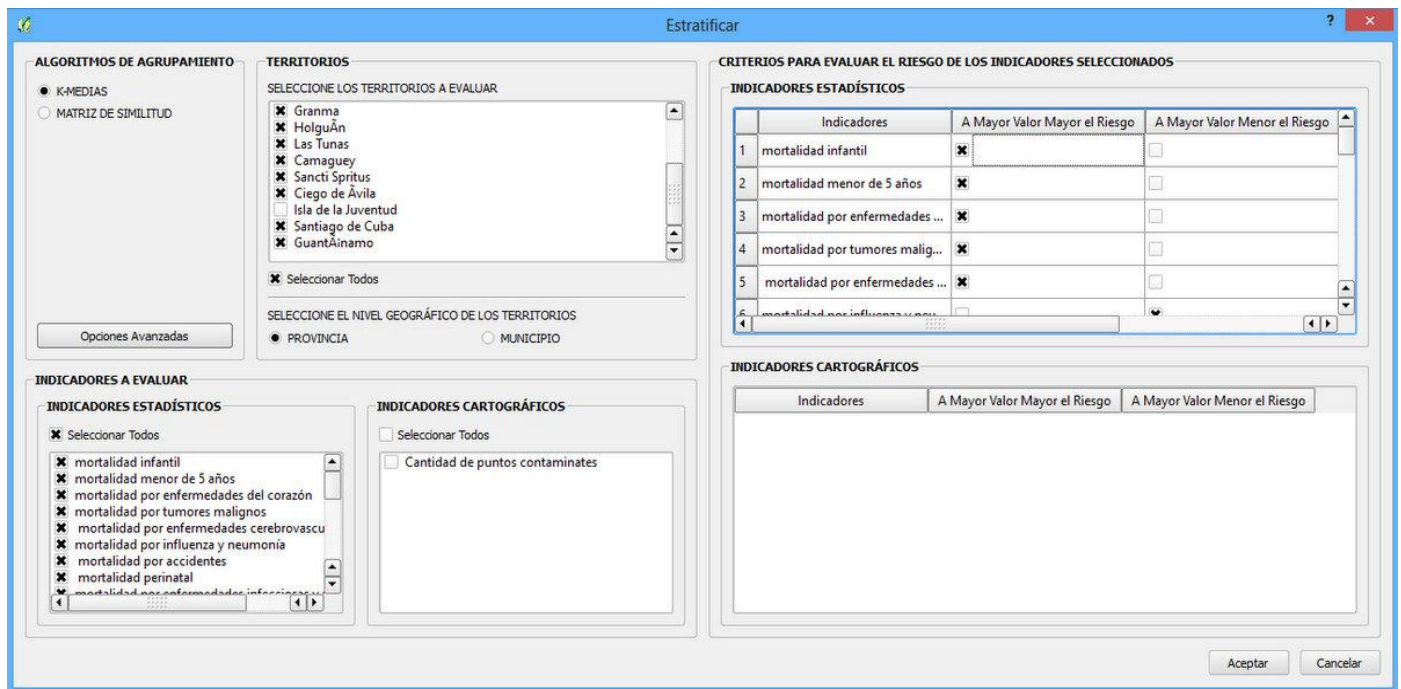


Figura 11. Interfaz de usuario VistaEstratificacion.

### 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 13](#) se aprecian los mismos de forma mapificada y en la [tabla 17](#) de manera más detallada.

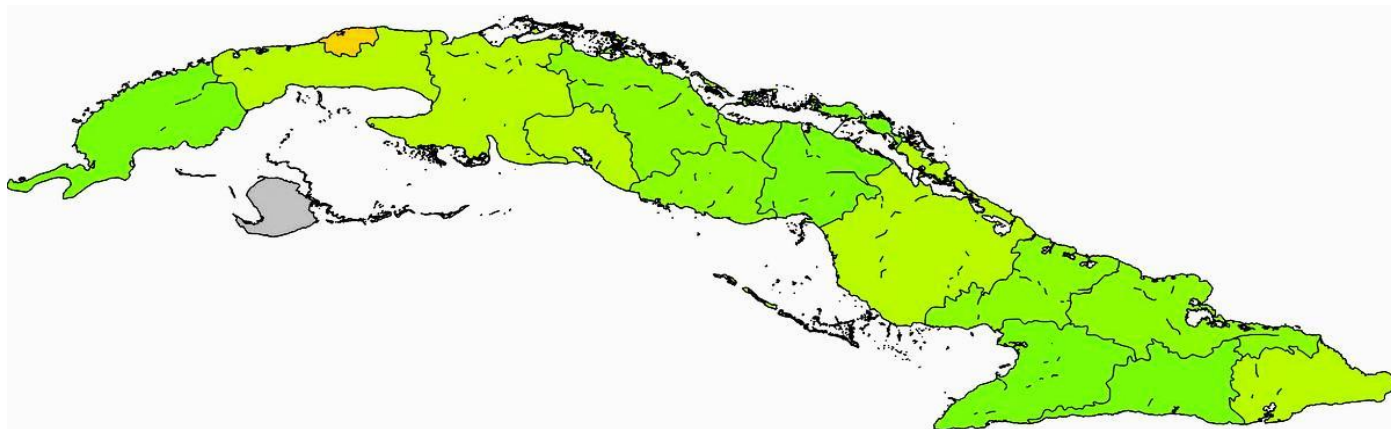


Figura 12. Mapa temático de la estratificación realizada utilizando la herramienta propuesta.

Nombre del estrato	Provincias	Riesgo de salud
Estrato 1	Ciudad de la Habana	0.76665
Estrato 2	Holguín, Las Tunas, Villa Clara, Sancti Spíritus	0.47898
Estrato 3	Matanzas, Cienfuegos, La Habana, Camagüey, Guantánamo	0.51907
Estrato 4	Pinar del Río, Santiago de Cuba, Ciego de Ávila, Granma	0.46165

Tabla 17. Resultados de la estratificación realizada utilizando la herramienta propuesta.

### Comparación de resultados

Para determinar la similitud existente entre los resultados obtenidos por la herramienta desarrollada y los del proceso realizado en (Yenisei Bombino Companioni 2005), a partir del caso de estudio descrito anteriormente, se utilizó la medida para evaluar la exactitud de los estratos resultantes que se describe a continuación.

Suponiendo que el número final de estratos es  $k$ , la medida de exactitud  $r$  está dada por la ecuación siguiente (Edna 2006).

$$r = \frac{\sum_{i=1}^k a_i}{n}$$

donde  $n$  es el número de territorios de la estratificación,  $a_i$  es el número de territorios que aparecen clasificados correctamente en el estrato  $i$  y en su correspondiente clase, la cual es aquella que tenga el número máximo. En otras palabras,  $a_i$  es el número de territorios con las etiquetas de la clase que dominan en el estrato  $i$ . Por lo tanto, el error está definido en la ecuación

$$e = 1 - r$$

Una medida de error en un rango  $\leq 0.3$ , representa un valor aceptable de error (Edna 2006).

A continuación se presenta una tabla comparativa entre los resultados del proceso de estratificación realizado utilizando la herramienta desarrollada y el efectuado en (Yenisei Bombino Companioni 2005).

<b>Resultados del proceso realizado en (Yenisei Bombino Companioni 2005)</b>	<b>Resultados obtenidos utilizando la herramienta desarrollada</b>
Cienfuegos, Santiago de Cuba, Ciego de Ávila	Pinar del Río, Santiago de Cuba, Ciego de Ávila, Granma
Pinar del Río, La Habana, Matanzas, Sancti Spíritus, Camagüey, Guantánamo	Matanzas, Cienfuegos, La Habana, Camagüey, Guantánamo
Ciudad de la Habana	Ciudad de la Habana
Holguín, Las Tunas, Villa Clara, Granma	Holguín, Las Tunas, Villa Clara, Sancti Spíritus

Tabla 18. Tabla comparativa de los resultados de los procesos de estratificación realizados (por estratos).

A partir de los resultados obtenidos en ambos procesos de estratificación, se aplica la medida de exactitud descrita inicialmente, obteniéndose un error de 0.28571.

Teniendo en cuenta lo anterior se evidencia que existe un valor del error aceptable entre los resultados obtenidos en ambos procesos. Esto permite verificar la veracidad de los estratos generados por la herramienta desarrollada a partir del caso de estudio propuesto.

Los resultados expuestos demuestran la utilidad y validez del empleo de la estratificación territorial, permitiendo obtener una valiosa información estratificada de la situación salud-enfermedad del país, posibilitando implementar acciones más efectivas, y facilitando la distribución de los recursos con un enfoque equitativo.

### **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. El estándar de codificación utilizado proporcionó un buen entendimiento del código y una mejor organización del mismo. 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. El valor de la medida del error demostró que existe un grado de similitud aceptable entre los procesos descritos. La solución desarrollada posibilitó realizar análisis de los resultados obtenidos en el proceso de estratificación de las provincias de Cuba, identificando cuales de estas presentan mayor riesgo de salud. Al concluir el período de pruebas se obtuvo una aplicación que cumple de forma correcta con la totalidad de las funcionalidades esperadas por el cliente.

## **CONCLUSIONES GENERALES**

Como resultados de la presente investigación se obtuvo una propuesta de solución para 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 definición del marco teórico referencial de la investigación relacionado con el proceso de estratificación de territorios basado en indicadores de salud, fundamentaron la necesidad de desarrollar un plugin que se adapte a los objetivos expuestos y satisfaga las necesidades del país.
- La revisión del panorama actual de los softwares SIG permitió la selección de QGIS para la integración de la solución propuesta.
- La integración de la solución propuesta al sistema QGIS facilitó la realización del proceso de estratificación de territorios utilizando indicadores de variada naturaleza.
- Las pruebas aplicadas para la verificación 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 a la herramienta desarrollada la funcionalidad de efectuar análisis estadísticos para realizar comparaciones entre los resultados de las estratificaciones obtenidas.
- Incorporar otros algoritmos de agrupamiento a la solución propuesta que faciliten el análisis de la componente espacial de los datos.



## REFERENCIAS BIBLIOGRÁFICAS

ALBA-CABRERA, E., 1997, *Nuevas extensiones del concepto de testor para diferentes tipos de funciones de semejanza*. Tesis para obtener el grado de Doctor en Ciencias Matemáticas, ICIMAF, Cuba.

BATISTA MOLINER, Ricardo, FEAL CAÑIZARES, Pablo, COUTIN MARIE, G., RODRÍGUEZ MILORD, D. and GONZÁLEZ CRUZ, R., 2001, *Guía para la realización del proceso de estratificación epidemiológica*. La Habana: MINSAP. 2001.

BECK, Kent, 2000, *Extreme programming explained: embrace change* [online]. Addison-Wesley Professional. [Accessed 20 May 2015]. Available from: <http://books.google.es/books?hl=es&lr=&id=G8EL4H4vf7UC&oi=fnd&pg=PR13&dq=Extreme+Programing+Explained&ots=j9vFtsgXyl&sig=Xz6T5Ne01wTeLnPskTctYLBSTdo>

BERZAL, Fernando, 2004, *Clustering basado en particiones*. [online]. 2004. [Accessed 20 May 2015]. Available from: <http://elvex.ugr.es/idbis/dm/slides/41%20Clustering%20-%20Partitional.pdf>

BRAVO, J. Domínguez, 2000, *Breve introducción a la cartografía ya los sistemas de información geográfica (SIG)* [online]. Ciemat. [Accessed 20 May 2015]. Available from: <http://hc.rediris.es/pub/bscw.cgi/d251342/itcsig.pdf>

CASAS, Sandra and REINAGA, Héctor, 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>

COBO, Ángel, 2007, *Diseño y programación de bases de datos* [online]. Editorial Visión Libros. [Accessed 20 May 2015]. Available from: <http://books.google.es/books?hl=es&lr=&id=anCDr9N-kGsC&oi=fnd&pg=PA7&dq=Dise%C3%B1o+y+programaci%C3%B3n+de+bases+de+datos&ots=UXEBp8mpzV&sig=jPWxCyBUit3XHIQlr4NpzhIbUwQ>

CRAIG, Larman, 1999, *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. Prentice Hall, Hispanoamérica México. 1999.

Cuba enfrenta la contaminación ambiental | Isla al Sur, 2015. [online], [Accessed 1 June 2015]. Available from: <http://islalsur.blogia.com/2013/072704-cuba-enfrenta-la-contaminacion-ambiental.php>

CUBAEDUCA, 2015, Portal Educativo Cubano - Biología. *Portal Educativo Cubano* [online]. 2015. [Accessed 1 June 2015]. Available from: [http://biologia.cubaeduca.cu/index.php?option=com\\_content&view=article&id=11328%3Atema9no-gradobiologia-salud-ambiental](http://biologia.cubaeduca.cu/index.php?option=com_content&view=article&id=11328%3Atema9no-gradobiologia-salud-ambiental)

DUQUE, Raúl González, 2011, *Python para todos*. [online]. 2011. [Accessed 20 May 2015]. Available

- from: <http://dspace.universia.net/handle/2024/919>
- EDNA, Hernández Valadez, 2006, Algoritmo de clustering basado en entropía para descubrir grupos en atributos de tipo mixto. [online]. 2006. [Accessed 20 May 2015]. Available from: <http://webserver.cs.cinvestav.mx/TesisGraduados/2006/tesisEdnaHernandez.pdf>
- ENTORNOS DE PROGRAMACIÓN, 2012, Entornos de programación. [online]. 2012. [Accessed 21 May 2015]. Available from: <http://lml.ls.fi.upm.es/ep/entornos.html#toc5>
- ERIC RODRIGUEZ, 2010, Patrones gof. [online]. 2010. [Accessed 20 May 2015]. Available from: [http://www.academia.edu/5903473/Patrones\\_gof](http://www.academia.edu/5903473/Patrones_gof)
- ERIK LIMÓN, 2012, tutorial estratificador INEGI - Buscar con Google. *Instituto Nacional de Estadística y Geografía (INEGI)* [online]. February 2012. [Accessed 20 May 2015]. Available from: [https://www.google.com.cu/?gws\\_rd=cr,ssl&ei=noEsVf65lvaAsQSatoCYDQ#q=tutorial+estratificador+INEGI](https://www.google.com.cu/?gws_rd=cr,ssl&ei=noEsVf65lvaAsQSatoCYDQ#q=tutorial+estratificador+INEGI)
- FRANCISCO JOSÉ CORTIJO BON, 2001, *Técnicas no supervisadas: Métodos de agrupamiento Bon Francisco Cortijo - Buscar con Google* [online]. [Accessed 20 May 2015]. Available from: [https://www.google.com.cu/?gws\\_rd=cr,ssl&ei=noEsVf65lvaAsQSatoCYDQ#q=Técnicas+no+supervisadas:+M%C3%A9todos+de+agrupamiento+Bon+Francisco+Cortijo](https://www.google.com.cu/?gws_rd=cr,ssl&ei=noEsVf65lvaAsQSatoCYDQ#q=Técnicas+no+supervisadas:+M%C3%A9todos+de+agrupamiento+Bon+Francisco+Cortijo)
- GONZÁLEZ, Damaris Pascual, 2010, *Algoritmos de agrupamiento basados en densidad y variación de clusters* [online]. Universitat Jaume I, Departament de Llenguatges i Sistemes Informàtics. [Accessed 20 May 2015]. Available from: <http://www.cerpamid.co.cu/sitio/files/DamarisTesis.pdf>
- GRASS DEVELOPMENT TEAM, 1998, GRASS GIS - Home. *The world's leading Free GIS software* [online]. 2015 1998. [Accessed 20 May 2015]. Available from: <http://grass.osgeo.org/>
- INFOMED, 2001, Anuario Estadístico de Cuba | Biblioteca Virtual en Salud de Cuba. [online]. 2001. [Accessed 20 May 2015]. Available from: <http://bvscuba.sld.cu/anuario-estadistico-de-cuba/>
- J. BÖHNER, 2015, SAGA - System for Automated Geoscientific Analyses. *System for Automated Geoscientific Analyses* [online]. 2015. [Accessed 20 May 2015]. Available from: <http://www.saga-gis.org/en/index.html>
- JETBRAINS INC, 2014, Python IDE & Django IDE for Web developers : JetBrains PyCharm. [online]. 2014. [Accessed 20 May 2015]. Available from: <https://www.jetbrains.com/pycharm/>
- JOSKOWICZ, José, 2008, Reglas y prácticas en eXtreme Programming. *Universidad de Vigo*. 2008. P. 22.
- JUAN PELÁEZ, 2009, Arquitectura basada en capas. [online]. May 2009. [Accessed 20 May 2015]. Available from: <http://www.juanpelaez.com/geek-stuff/arquitectura/arquitectura-basada-en-capas/>
- LARMAN, Craig, 1999, *UML y Patrones* [online]. Pearson. [Accessed 21 May 2015]. Available from:

- <http://gravepa.com/granaino/biblioteca/aprende/UNED%20-%20Grado%20Inform%C3%A1tica%20-%20Extras%20%26%20Ediciones%20Antiguas/Una%20introduccion%20al%20analisis%20y%20diseno%20orientado%20a%20objetos%20y%20al%20proceso%20unificado.pdf>
- LETELIER, Patricio, 2006, Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [online]. 2006. [Accessed 20 May 2015]. Available from: [http://www.cyta.com.ar/ta0502/b\\_v5n2a1.htm](http://www.cyta.com.ar/ta0502/b_v5n2a1.htm)
- LÓPEZ CAVIEDES, Marco Antonio, 2004, Herramienta para la estratificación de municipios en zonas de riesgo para la salud. [online]. 2004. [Accessed 20 May 2015]. Available from: <http://dgsa.uaeh.edu.mx:8080/xmlui/handle/123456789/29>
- LORENZO MARTÍNEZ, RM, 2000, Cartografía, urbanismo y desarrollo inmobiliario. *Dossat*. 2000.
- LOUDEN, Kenneth C., 2004, *Lenguajes de programación: Principios y práctica*. Cengage Learning Latin America.
- MIGUEL ANGEL GONZALEZ, 2012, HERRAMIENTAS CASE. [online]. 13 September 2012. [Accessed 20 May 2015]. Available from: [https://prezi.com/aad8mbta\\_vjb/herramientas-case/](https://prezi.com/aad8mbta_vjb/herramientas-case/)
- MOLINA LÓPEZ, J. M. and GARCÍA HERRERO, J., 2006, *Técnicas de análisis de datos. Aplicaciones Prácticas utilizando Microsoft Excel y WEKA*. Madrid, Universidad Carlos III.
- ORTAG, Felix, 2015, Map production. *International Cartographic Association* [online]. February 2015. [Accessed 20 May 2015]. Available from: <http://icaci.org/research-agenda/map-production/> Map production, Map production
- PASCUAL, D., PLA, F. and SÁNCHEZ, S., 2007, Algoritmos de agrupamiento. *Método Informáticos Avanzados* [online]. 2007. [Accessed 20 May 2015]. Available from: [http://marmota.dlsi.uji.es/WebBIB/papers/2007/1\\_Pascual-MIA-2007.pdf](http://marmota.dlsi.uji.es/WebBIB/papers/2007/1_Pascual-MIA-2007.pdf)
- POSTGIS DEVELOPMENT TEAM, 2014, PostGIS. [online]. 10 September 2014. [Accessed 20 May 2015]. Available from: <http://postgis.refractor.net/>
- POSTGRESQL-3 GLOBAL DEVELOPMENT GROUP, 2014, PostgreSQL: Documentation: 9.0: Release 9.0.1. [online]. 2014. [Accessed 20 May 2015]. Available from: <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 I McGraw-Hill. Interamericana Editores, SA de CV México.
- QGIS DEVELOPMENT TEAM, 2012, QGIS project! *QGIS A Free and Open Source Geographic Information System* [online]. 2012. [Accessed 20 May 2015]. Available from: <http://www.qgis.org/en/site/>

- ROBINSON, Christina, 2011, Basic introduction into pgAdmin III and SQL queries. [online]. 2011. [Accessed 20 May 2015]. Available from: <http://library.thehumanjourney.net/658/>
- RODRÍGUEZ, Jorge Enrique Rodríguez, BLANCO, Edwar Alonso Rojas and CAMACHO, Roger Orlando Franco, 2013, Clasificación de datos usando el método k-nn. *Vínculos*. 2013. Vol. 4, no. 1, p. 4–18.
- ROJAS, Luisa Iñiguez, 1998, Geografía y salud: temas y perspectivas en América Latina Geography and health: themes and perspectives in Latin America. *Cad. Saúde Pública*. 1998. Vol. 14, no. 4, p. 701–711.
- SCHEFER-WENZL, Sigrid, SOBERNIG, Stefan and STREMBECK, Mark, 2013, Evaluating A Uml-Based Modeling Framework For Process-Related Security Properties: A Qualitative Multi-Method Study. In : *ECIS* [online]. 2013. p. 134. [Accessed 20 May 2015]. Available from: [http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1357&context=ecis2013\\_cr](http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1357&context=ecis2013_cr)
- SOMMERVILLE, Ian, 2005, *Ingeniería del software* [online]. Pearson Educación. [Accessed 20 May 2015]. Available from: <http://books.google.es/books?hl=es&lr=&id=gQWd49zSut4C&oi=fnd&pg=PA1&dq=Ingenieria+de+Software+Ian+Sommerville&ots=s623rrszwd&sig=DVAJiDo0wOzveatAqj48nPXvtsl>
- STARTED, GETTING, 2010, Software Design Tools for Agile Teams, with UML, BPMN and More. [online]. 2010. [Accessed 20 May 2015]. Available from: <http://www.visual-paradigm.com/>
- TORRES, Alexander Rodríguez and PUENTE, Rafael Rodríguez, 2010, Servicio de mapas temáticos. *Mapping*. 2010. No. 139, p. 36–39.
- VÍCTOR, Olaya, 2011, Sistemas de Información Geográfica. *Libro SIG* [online]. 2011. Vol. 1.0. Available from: [http://wiki.osgeo.org/wiki/Libro\\_SIG](http://wiki.osgeo.org/wiki/Libro_SIG)
- XU, Rui and WUNSCH, Don, 2008, *Clustering* [online]. John Wiley & Sons. [Accessed 20 May 2015]. Available from: [http://books.google.es/books?hl=es&lr=&id=kYC3YCYl\\_tkC&oi=fnd&pg=PR5&dq=Rui+Xu+Clustering&ots=qj83yEbjXD&sig=UuJh9rLViUwClqxmUQSdkIS0BJ4](http://books.google.es/books?hl=es&lr=&id=kYC3YCYl_tkC&oi=fnd&pg=PR5&dq=Rui+Xu+Clustering&ots=qj83yEbjXD&sig=UuJh9rLViUwClqxmUQSdkIS0BJ4)
- YENISEI BOMBINO COMPANIONI, 2005, RTV-Tipología Estructural Articulada. *metodología de análisis para la estratificación según indicadores de salud* [online]. 14 November 2005. [Accessed 20 May 2015]. Available from: [http://bvs.sld.cu/uats/rtv\\_files/2005/bombino.htm](http://bvs.sld.cu/uats/rtv_files/2005/bombino.htm)

## GLOSARIO DE TÉRMINOS

**CASE** herramienta que brinda asistencia a los analistas (del inglés Computer Aided Software Engineering)

**CRC** Clase, Responsabilidad y Colaboración (del inglés *Class, Responsibility and Collaboration*)

**DER** Diagrama Entidad-Relación

**GoF** Patrones del Grupo de Cuatro (del inglés *Gang of Four*)

**GRASS** Sistema de Apoyo al Análisis de los Recursos Geográficos (del inglés *Geographic Resources Analysis Support System*)

**GRASP** Patrones Generales de Software para Asignación de Responsabilidades (del inglés *General Responsibility Assignment Software Patterns*)

**HU** Historia de Usuario (artefacto generado por la metodología Programación Extrema)

**MST** Mínimo Árbol de Expansión (del inglés *Minimum Spanning Tree*)

**QGIS** Quantum GIS (Software de Información geográfica)

**RF** Requisito Funcional

**RLCP** Reconocimiento Lógico Combinatorio de Patrones

**RNF** Requisito No Funcional

**SAGA** Sistema de Análisis Geocientífico Automatizado (del inglés *System for Automated Geoscientific Analyses*)

**SIG** Sistema de Información Geográfica.

**XP** Metodología de desarrollo de software ágil (del inglés *Extreme Programing*)

## ANEXOS

## Anexo1. Historias de Usuario

Historia de Usuario “Importar indicadores estadísticos desde una hoja de cálculo”	
<b>Número:</b> 1	<b>Nombre Historia de Usuario:</b> Importar indicadores estadísticos desde una hoja de cálculo.
<b>Usuario:</b> Experto	
<b>Prioridad en Negocio:</b> Media	<b>Riesgo en Desarrollo:</b> Media
<b>Puntos Estimados:</b> 7	<b>Iteración Asignada:</b> 4
<b>Programador responsable:</b> Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> La aplicación debe ser capaz de obtener los indicadores estadísticos de los territorios a evaluar desde una hoja de cálculo seleccionada por el usuario.	
<b>Observaciones:</b>	

Historia de Usuario “Obtener características cartográficas a través de QGIS	
<b>Número:</b> 2	<b>Nombre Historia de Usuario:</b> Obtener características cartográficas a través de QGIS.
<b>Usuario:</b> Experto	
<b>Prioridad en Negocio:</b> Medio	<b>Riesgo en Desarrollo:</b> Medio
<b>Puntos Estimados:</b> 7	<b>Iteración Asignada:</b> 4
<b>Programador responsable:</b> Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> La aplicación debe ser capaz de capturar la información necesaria del SIG QGIS para construir una estratificación, así como la capa correspondiente de los territorios a evaluar; y si es preciso, las capas necesarias para el cálculo de los valores de los indicadores cartográficos: “cantidad de ríos con contaminación” y “cantidad de puntos contaminantes”.	
<b>Observaciones:</b>	

Historia de Usuario “Construir estratos”	
Número: 3.1	Nombre Historia de Usuario: Construir estratos
Usuario: Experto	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos Estimados: 21	Iteración Asignada: 1
Programador responsable: Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> La aplicación debe ser capaz de construir estratos, a partir de: <ul style="list-style-type: none"> <li>• los indicadores estadísticos importados por el usuario desde una hoja de cálculo.</li> <li>• las características cartográficas seleccionadas por el usuario a través de QGIS.</li> </ul>	
<b>Observaciones:</b>	

Historia de Usuario “Visualizar estratos construidos en mapa temático”	
Número: 3.2	Nombre Historia de Usuario: Visualizar estratos construidos en mapa temático.
Usuario: Experto	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos Estimados: 21	Iteración Asignada: 2
Programador responsable: Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> La aplicación debe ser capaz de representar gráficamente en un mapa temático cada estratificación generada.	
<b>Observaciones:</b>	

Historia de Usuario “Adicionar estratificación”	
Número: 4.1	Nombre Historia de Usuario: Adicionar estratificación.
Usuario: Experto	

<b>Prioridad en Negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Alto
<b>Puntos Estimados:</b> 3	<b>Iteración Asignada:</b> 3
<b>Programador responsable:</b> Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> La aplicación debe ser capaz de almacenar en una base de datos cada estratificación generada.	
<b>Observaciones:</b>	

Historia de Usuario “Mostrar estratificación”	
<b>Número:</b> 4.2	<b>Nombre Historia de Usuario:</b> Mostar estratificación.
<b>Usuario:</b> Experto	
<b>Prioridad en Negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Alto
<b>Puntos Estimados:</b> 3	<b>Iteración Asignada:</b> 3
<b>Programador responsable:</b> Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> La aplicación debe ser capaz de representar en un mapa temático los resultados de cada estratificación almacenada en la base de datos.	
<b>Observaciones:</b>	

Historia de Usuario “Eliminar estratificación”	
<b>Número:</b> 4.3	<b>Nombre Historia de Usuario:</b> Eliminar estratificación
<b>Usuario:</b> Experto	
<b>Prioridad en Negocio:</b> Alto	<b>Riesgo en Desarrollo:</b> Alto
<b>Puntos Estimados:</b> 3	<b>Iteración Asignada:</b> 3
<b>Programador responsable:</b> Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> La aplicación debe ser capaz de eliminar cada estratificación almacenada en la base de datos.	
<b>Observaciones:</b>	



Historia de Usuario “Exportar mapa temático de una estratificación como imagen”	
Número: 5	Nombre Historia de Usuario: Exportar mapa temático de una estratificación como imagen.
Usuario: Experto	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos Estimados: 7	Iteración Asignada: 5
Programador responsable: Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> La aplicación debe ser capaz de exportar como imagen el mapa temático cada estratificación generada.	
<b>Observaciones:</b>	

Historia de Usuario “Exportar estratificación hacia una hoja de cálculo”	
Número: 6	Nombre Historia de Usuario: Exportar estratificación hacia una hoja de cálculo.
Usuario: Experto	
Prioridad en Negocio: Alto	Riesgo en Desarrollo: Alto
Puntos Estimados: 7	Iteración Asignada: 4
Programador responsable: Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> La aplicación debe ser capaz de exportar los resultados de cada estratificación generada hacia una hoja de cálculo.	
<b>Observaciones:</b>	

#### Anexo2. Tareas de ingeniería asociadas a la HU # 3.1.

Tarea de ingeniería	
Número Tarea: 1	Número Historia de Usuario: HU # 3.1

<b>Nombre Tarea:</b> Agrupar territorios utilizando el algoritmo k-medias.	
<b>Tipo Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 16/02/2015	<b>Fecha Fin:</b> 20/02/2015
<b>Programador Responsable:</b> Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> Esta tarea permite agrupar los territorios seleccionados por el utilizando el algoritmo k-medias.	

Tarea de ingeniería	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> HU # 3.1
<b>Nombre Tarea:</b> Agrupar territorios utilizando el algoritmo basado en la matriz de similitud.	
<b>Tipo Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 1
<b>Fecha Inicio:</b> 23/02/2015	<b>Fecha Fin:</b> 27/02/2015
<b>Programador Responsable:</b> Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> Esta tarea permite agrupar los territorios seleccionados por el usuario utilizando el algoritmo basado en la matriz de similitud.	

Tarea de ingeniería	
<b>Número Tarea:</b> 3	<b>Número Historia de Usuario:</b> HU # 3.1
<b>Nombre Tarea:</b> Obtener el aporte informacional de los indicadores.	
<b>Tipo Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 2/03/2015	<b>Fecha Fin:</b> 3/03/2015
<b>Programador Responsable:</b> Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> Esta tarea permite obtener el aporte informacional de cada uno de los indicadores seleccionados por el usuario.	

Tarea de ingeniería	
<b>Número Tarea:</b> 4	<b>Número Historia de Usuario:</b> HU # 3.1
<b>Nombre Tarea:</b> Obtener el aporte de riesgo de salud de los territorios.	
<b>Tipo Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 4/03/2015	<b>Fecha Fin:</b> 5/03/2015
<b>Programador Responsable:</b> Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> Esta tarea permite obtener el aporte de riesgo de salud cada uno de los territorios seleccionados por el usuario.	

Tarea de ingeniería	
<b>Número Tarea:</b> 5	<b>Número Historia de Usuario:</b> HU # 3.1
<b>Nombre Tarea:</b> Normalizar los datos de los indicadores.	
<b>Tipo Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.3
<b>Fecha Inicio:</b> 5/03/2015	<b>Fecha Fin:</b> 6/03/2015
<b>Programador Responsable:</b> Rolando Morales Pérez, Yanislay Torres Vega	
<b>Descripción:</b> Esta tarea permite normalizar los datos de cada uno de los indicadores seleccionados por el usuario.	

### Anexo3. Tarjetas CRC

Clase: ControladorEstratificador	
Responsabilidades	Colaboración
<ul style="list-style-type: none"> <li>Controlar las estratificaciones generadas.</li> <li>Construir estratificaciones.</li> </ul>	FuenteDatos, VistaPrincipal, Estratificacion, Visualizacion, Algoritmo, CalculoIndicadoresCartograficos

<b>Clase: Territorio</b>	
Responsabilidades	Colaboración
<ul style="list-style-type: none"> <li>• Calcular el aporte de riesgo los territorios.</li> <li>• Crear instancias de la clase Indicador.</li> </ul>	Estrato, Indicador, Algoritmo

<b>Clase: Estratificación</b>	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Crear instancias de la clase Estrato.</li> </ul>	ControladorEstratificador, Estrato

<b>Clase: Indicador</b>	
Responsabilidades	Colaboración
<ul style="list-style-type: none"> <li>• Calcular el aporte de riesgo los indicadores.</li> <li>• Normalizar los valores de cada indicador.</li> <li>• Calcular el aporte informacional de los indicadores.</li> </ul>	Territorio

<b>Clase: Estrato</b>	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Crear instancias de la clase Territorio.</li> <li>• Calcular el aporte de riesgo de cada estrato.</li> </ul>	Estratificación, Territorio

<b>Clase: Visualización</b>	
Responsabilidades	Colaboración

<ul style="list-style-type: none"> <li>• Crear una capa para cada estratificación generada.</li> <li>• Asignar los colores correspondientes a cada territorio en base al aporte de riesgo de su estrato.</li> </ul>	ControladorEstratificador
---	---------------------------

Clase: AlgoritmoKMedias	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Agrupar los territorios utilizando el algoritmo k-medias.</li> </ul>	Algoritmo

Clase: AlgoritmoMatrizSimilitud	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Agrupar los territorios utilizando el algoritmo de agrupamiento basado en la matriz de similitud.</li> </ul>	Algoritmo

Clase: CalculoIndicadoresCartograficos	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>• Calcular los valores del indicador <i>cantidad de ríos con contaminación</i>.</li> <li>• Calcular los valores del indicador <i>cantidad de puntos contaminantes</i>.</li> </ul>	ControladorEstratificador

Clase: FuenteDatosPostgis	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>Gestionar la base de datos para las estratificaciones almacenadas.</li> </ul>	FuenteDatos

Clase: FuenteDatosExcel	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> <li>Capturar los indicadores estadísticos de los territorios a evaluar desde una hoja de cálculo.</li> <li>Exportar los resultados de cada estratificación generada hacia una hoja de cálculo.</li> </ul>	FuenteDatos

#### Anexo4. Casos de prueba de aceptación

Caso de prueba de aceptación	
Código: HU1_P1	Historia de Usuario: 1
<b>Nombre:</b> Importar indicadores estadísticos desde una hoja de cálculo	
<b>Descripción:</b> Prueba para validar la funcionalidad importar indicadores estadísticos desde una hoja de cálculo.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"> <li>El usuario debe seleccionar la opción <i>Importar datos estadísticos desde una hoja de cálculo</i>.</li> <li>El usuario debe seleccionar la hoja de cálculo que desea importar.</li> <li>El usuario debe seleccionar la opción <i>Abrir</i>.</li> </ul>	
<b>Resultados esperados:</b> En caso que la hoja de cálculo sea válida, el sistema muestra un mensaje informando que los datos han sido importados satisfactoriamente. En caso contrario el sistema	

muestra un mensaje anunciando que los datos importados no son válidos.
<b>Evaluación de la prueba:</b> Prueba satisfactoria

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU3.2_P1	<b>Historia de Usuario:</b> 3.2
<b>Nombre:</b> Visualizar estratos contruidos en mapa temático.	
<b>Descripción:</b> Prueba para validar la funcionalidad visualizar estratos contruidos en mapa temático.	
<b>Condiciones de ejecución:</b>	
<ul style="list-style-type: none"> <li>• El sistema debe haber construido los estratos.</li> </ul>	
<b>Resultados esperados:</b> El sistema muestra los estratos contruidos en un mapa temático.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria	

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU4.1_P1	<b>Historia de Usuario:</b> 4.1
<b>Nombre:</b> Adicionar estratificación.	
<b>Descripción:</b> Prueba para validar la funcionalidad adicionar estratificación.	
<b>Condiciones de ejecución:</b>	
<ul style="list-style-type: none"> <li>• El usuario debe haber construido al menos una estratificación.</li> <li>• El usuario debe conectarse a la base de datos.</li> <li>• El usuario debe definir una descripción para la estratificación que desea adicionar.</li> <li>• El usuario debe seleccionar la opción <i>Adicionar estratificación</i>.</li> </ul>	
<b>Resultados esperados:</b> En caso que se cumplan las condiciones de ejecución, el sistema muestra un mensaje informando que la estratificación se ha añadido correctamente. En caso contrario el sistema muestra un mensaje anunciando el motivo por el cual no adicionó la estratificación a la base de datos.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria	

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU4.2_P1	<b>Historia de Usuario:</b> 4.2
<b>Nombre:</b> Mostrar estratificación.	
<b>Descripción:</b> Prueba para validar la funcionalidad mostrar estratificación.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"> <li>• El usuario debe conectarse a la base de datos.</li> <li>• El usuario debe seleccionar una estratificación.</li> <li>• El usuario debe seleccionar la opción <i>Visualizar estratificación</i>.</li> </ul>	
<b>Resultados esperados:</b> En caso que se cumplan las condiciones de ejecución, el sistema muestra la estratificación seleccionada en un mapa temático. En caso contrario el sistema muestra un mensaje informando el motivo por el cual no se muestra la estratificación seleccionada.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria	

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU4.3_P1	<b>Historia de Usuario:</b> 4.3
<b>Nombre:</b> Eliminar estratificación.	
<b>Descripción:</b> Prueba para validar la funcionalidad eliminar estratificación.	
<b>Condiciones de ejecución:</b> <ul style="list-style-type: none"> <li>• El usuario debe conectarse a la base de datos.</li> <li>• El usuario debe seleccionar una estratificación.</li> <li>• El usuario debe seleccionar la opción <i>Eliminar estratificación</i></li> </ul>	
<b>Resultados esperados:</b> En caso que se cumplan las condiciones de ejecución, el sistema muestra un mensaje informando que la estratificación seleccionada se ha eliminado correctamente. En caso contrario el sistema muestra un mensaje informando el motivo por el cual no eliminó la estratificación.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria	



<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU5_P1	<b>Historia de Usuario:</b> 5
<b>Nombre:</b> Exportar mapa temático de cada estratificación como imagen.	
<b>Descripción:</b> Prueba para validar la funcionalidad exportar mapa temático de cada estratificación como imagen.	
<b>Condiciones de ejecución:</b>	
<ul style="list-style-type: none"> <li>• El usuario debe haber construido al menos una estratificación.</li> <li>• El usuario debe seleccionar la opción <i>Guardar imagen del mapa de la estratificación actual</i>.</li> <li>• El usuario debe ubicarse en el lugar donde desea guardar la imagen.</li> <li>• El usuario debe definir un nombre para la imagen que se desea guardar.</li> <li>• El usuario debe seleccionar la opción <i>Guardar</i>.</li> </ul>	
<b>Resultados esperados:</b> En caso que se cumplan las condiciones de ejecución, el sistema guarda la imagen. En caso contrario el sistema muestra un mensaje informando el motivo por el cual no se guardó la imagen.	
<b>Evaluación de la prueba:</b> Prueba satisfactoria	

<b>Caso de prueba de aceptación</b>	
<b>Código:</b> HU6_P1	<b>Historia de Usuario:</b> 6
<b>Nombre:</b> Exportar estratificaciones hacia una hoja de cálculo.	
<b>Descripción:</b> Prueba para validar la funcionalidad exportar estratificaciones hacia una hoja de cálculo.	
<b>Condiciones de ejecución:</b>	
<ul style="list-style-type: none"> <li>• El usuario debe haber construido al menos una estratificación.</li> <li>• El usuario debe seleccionar la opción <i>Exportar resultados hacia una hoja de cálculo</i>.</li> <li>• El usuario debe ubicarse en el lugar donde desea exportar los resultados.</li> <li>• El usuario debe definir un nombre para la hoja de cálculo.</li> </ul>	

- El usuario debe seleccionar la opción *Guardar*.

**Resultados esperados:** En caso de que se seleccionen correctamente todas las opciones, el sistema exporta los resultados de la estratificación hacia una hoja de cálculo. En caso contrario el sistema muestra un mensaje informando el motivo por el cual no se exportaron los resultados.

**Evaluación de la prueba:** Prueba satisfactoria