

Universidad de las Ciencias Informáticas

Facultad 3

Departamento de Programación



Componente para la selección de características relevantes sobre el sistema de estratificación de territorios.

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

Autores: Lissette Castro Palacio

Rosmery Hernández Peña

Tutores: Ing. Yadian Guillermo Pérez Betancourt

Ing. Liset González Polanco

Co-Tutor: Ing. Raniel Gé Vaillant

La Habana, 26 de junio de 2017

“Año 59 de la Revolución”



“Seamos realistas y hagamos lo imposible”

Che

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis que lleva como título: “Componente para la selección de características relevantes sobre el sistema de estratificación de territorios.” y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presenta a los _ días del mes de _____ del año _____.

Lisette Castro Palacio

Autor

Rosmery Hernández Peña

Autor

Ing. Yadian Guillermo Pérez Betancourt

Tutor

Ing. Liset González Polanco

Tutor

Ing. Raniel Gé Vaillant

Co-Tutor

DEDICATORIA

A mi familia por apoyarme siempre, en especial a Mima y Yeni.

A mi papá por ser el hombre más importante en mi vida y deseo seguir siendo su orgullo.

A mi mamá por sentir en carne propia cada uno de mis sufrimientos y logros y apoyarme incondicionalmente en todas mis decisiones.

A mis abuelos Arquímedes Castro (papatuti) y Pedro Palacio (palacio) regalo por ser la primera nieta universitaria.

A mi abuela Manuela, aunque ella deseaba que fuera doctora, esto es un paso para serlo, pero en Ciencias Informáticas.

A mi abuela Graciela, que aunque no está entre nosotros, sé que desde el cielo me cuida y me acompaña siempre y sobre todo está muy orgullosa de mí.

Y a las personas que dudaron de mí y me dieron fuerzas para demostrarles que cuando te digan no se puede, decirles: “mira como lo hago”.

Lissette Castro Palacio

AGRADECIMIENTOS

Agradezco a Dios por acompañarme siempre y permitirme este triunfo.

A la Revolución Cubana, por brindarme la posibilidad de estudiar y superarme.

A mi mamá y papá por confiar en mí y apoyarme siempre.

*A mi abuela por ser una de las personas más importantes en mi vida y haber sido
el pilar fundamental de este logro.*

A mis tías Ivian y Maritza por su apoyo incondicional y sus consejos

*A Nury por darme su apoyo y ser mi ejemplo a seguir, en fin por ser mi amiga y
hermana.*

*A los profesores que ayudaron de una forma u otra en mi formación como
profesional, en especial a Hugo Arnaldo Martínez Noriega por su apoyo
incondicional.*

*Al Ing. Yadian Guillermo Pérez Betancourt, mi tutor, por su excelente, impecable y
certera guía en la realización de este trabajo y por haber tenido la confianza en mí
de lograr este sueño.*

A todos mis compañeros que me apoyaron a lo largo de estos 5 años.

A mi hermana e inseparable, a pesar de la gran distancia, amigo Daliana.

Lisette Castro Palacio

Primeramente quiero agradecerle a dios por darme la bendición y el orgullo de crecer al lado de las personas más importantes en mi vida... mis padres.

A mi mamá por ser amiga y madre, por brindarme su apoyo incondicional, por enseñarme que en la vida todo es posible mientras se luche por lo que se quiere, por siempre estar pendiente de mí y guiarme por el mejor camino.

A mi padre por estar siempre a mi lado, por luchar siempre para que en mi rostro halla una sonrisa, porque a pesar de los momentos difíciles que ha tenido que transitar me demostró que no importa cuántas veces uno caiga lo importante es levantarse y saber seguir adelante.

A mis hermanos Yadrian, Yanosky y Yoan por confiar en mí y aconsejarme en las diferentes etapas de mi vida, y a mi hermano Chicho que aunque dios no le dio la oportunidad de estar por más tiempo en este mundo me enseñó que todo lo que uno quiere lo puede lograr y sobre todo que la vida es muy corta y es necesario disfrutar cada minuto.

A mi novio por estar conmigo en un momento tan especial, por enseñarme que la vida no se puede vivir de caprichos, por hacerme una persona más fuerte, gracias por acompañarme en los momentos difíciles y ser una motivación para seguir adelante.

A mi compañera de tesis, amiga y hermana, por estar siempre a mi lado en momentos que pensé que no podía seguir adelante.

A mi negrito Horsford, Ale, a todos los que he conocido en estos últimos tiempos que me han demostrado que puedo contar con ellos en todo momento.

A mi tutor por tener tanta paciencia conmigo y ayudarme a que todo esto fuera posible.

A todas las personas que a lo largo de estos cinco años han estado a mi lado, han compartido conmigo experiencias inolvidables, no quisiera mencionar nombres por temor a obviar a alguien, pero muchas gracias a todos por formar parte de mi vida.

Rosmery Hernández Peña.

RESUMEN

La mejora tecnológica en los sistemas de adquisición de datos, su abaratamiento y la creciente capacidad computacional de los ordenadores, facilita que la dimensionalidad de los datos y su almacenamiento crezca continuamente. Los SIG¹ constituyen una herramienta de gran importancia para los estudios relacionados con el espacio. Existen varios sectores que se nutren de ellos para favorecer el desempeño de sus tareas, tales como la agricultura, la planificación urbanista y la salud pública, solo por mencionar algunos.

Los SIG en el contexto de la salud son utilizados para el análisis de la distribución espacial de problemas de salud o de los servicios de la población. El análisis estratificado es una de las herramientas de análisis con mayor potencial en ese sector. Las soluciones existentes para la estratificación brindan la posibilidad a los usuarios de seleccionar las características o atributos de los territorios para realizar el proceso de forma manual. No permiten la selección automática de estas a partir de su relevancia, teniendo como resultado que las características sean tratadas todas por igual, aunque sean redundantes e irrelevantes.

El objetivo de la presente investigación es desarrollar un componente para la selección de características relevantes sobre el sistema de estratificación de territorios que facilite la obtención de información precisa en estudios salubristas, partiendo de un conjunto de técnicas y algoritmos para determinar un subconjunto mínimo de las características de un problema.

Palabras clave: algoritmos, atributos relevantes, estratificación, técnicas.

¹Significado para el término: Sistemas de Información Geográficos.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1. REFERENTES TEÓRICOS SOBRE LA ESTRATIFICACIÓN DE TERRITORIOS.....	6
1.1. SISTEMAS DE INFORMACIÓN GEOGRÁFICA.....	7
1.1.1. <i>Conceptos básicos asociados a los SIG.....</i>	<i>8</i>
1.1.2. <i>Panorama actual de aplicaciones SIG.....</i>	<i>8</i>
1.2. ANÁLISIS DE SOLUCIONES EXISTENTES CON SOPORTE PARA LA ESTRATIFICACIÓN	10
1.3. ATRIBUTOS RELEVANTES.....	13
1.3.1. <i>Enfoques y técnicas para la selección de atributos relevantes</i>	<i>14</i>
1.3.2. <i>Representación mediante un grafo del funcionamiento de las hormigas.....</i>	<i>15</i>
1.3.3. <i>Problema de clasificación.....</i>	<i>16</i>
1.4. HERRAMIENTAS, LENGUAJES Y TECNOLOGÍAS A UTILIZAR.....	18
1.4.1. <i>Lenguaje de modelado.....</i>	<i>18</i>
1.4.2. <i>Herramienta CASE</i>	<i>18</i>
1.4.3. <i>Lenguaje de programación</i>	<i>19</i>
1.4.4. <i>Gestor de base de datos.....</i>	<i>20</i>
1.4.5. <i>Metodología de desarrollo.....</i>	<i>22</i>
1.5. CONCLUSIONES PARCIALES	24
CAPÍTULO 2. COMPONENTE PARA LA SELECCIÓN DE CARACTERÍSTICAS RELEVANTES SOBRE EL SISTEMA DE ESTRATIFICACIÓN DE TERRITORIOS.....	25
2.1. SELECCIÓN DE ATRIBUTOS RELEVANTES	26
2.2. REQUISITOS DE SOFTWARE.....	30
2.3. REQUISITOS FUNCIONALES	30
2.4. REQUISITOS NO FUNCIONALES	30
2.5. FASE DE PLANIFICACIÓN.....	31
2.5.1. <i>Historias de usuarios.....</i>	<i>32</i>
2.5.2. <i>Estimación de esfuerzos por historias de usuario</i>	<i>33</i>
2.5.3. <i>Plan de iteraciones.....</i>	<i>34</i>

2.5.4.	<i>Plan de entrega.</i>	34
2.6.	FASE DE DISEÑO.	35
2.6.1.	<i>Tarjetas Clase-Responsabilidad-Colaboración.</i>	35
2.6.2.	<i>Arquitectura de software</i>	36
2.6.3.	<i>Estilo arquitectónico a utilizar</i>	36
2.7.	PATRONES DE DISEÑO	38
2.7.1.	<i>Patrones de diseño GRASP</i>	39
2.7.2.	<i>Patrones de diseño GoF.</i>	41
2.8.	CONCLUSIONES PARCIALES	42
CAPÍTULO 3.	VERIFICACIÓN DE LA SOLUCIÓN PROPUESTA	44
3.1.	FASE DE IMPLEMENTACIÓN	44
3.1.1.	<i>Tarea de ingeniería</i>	44
3.1.2.	<i>Estándares de codificación</i>	45
3.2.	FASE DE PRUEBAS	46
3.2.1.	<i>Pruebas de caja blanca</i>	46
3.2.2.	<i>Pruebas de aceptación</i>	51
3.2.3.	<i>Caso de estudio.</i>	52
3.3.	CONCLUSIONES PARCIALES	57
CONCLUSIONES		59
RECOMENDACIONES		60
REFERENCIAS BIBLIOGRÁFICAS		61

ÍNDICE DE FIGURAS

Figura 1: Modelo conceptual de la propuesta de solución.....	25
Figura 2: Evidencia de la arquitectura del sistema.	38
Figura 3: Evidencia del patrón experto.....	39
Figura 4: Evidencia del patrón creador.....	40
Figura 5: Evidencia del patrón controlador.....	40
Figura 6: Evidencia del patrón plantilla.....	42
Figura 7: Código del método cantidadDeHormigas ().....	47
Figura 8: Grafo de flujo del método cantidadDeHormigas ().....	47
Figura 9: Interfaz de usuario VistaEstratificador.....	54
Figura 10: Mapa temático de la estratificación realizada con la selección de características relevantes.	55

ÍNDICE DE TABLAS

Tabla 1: Comparación entre los sistemas analizados.	12
Tabla 2: Historia de usuario. Seleccionar características relevantes.	33
Tabla 3: Estimación del esfuerzo por historia de usuario.	33
Tabla 4: Plan de duración de iteraciones.	34
Tabla 5: Plan de entregas.	35
Tabla 6: Tarjeta CRC para la clase HormigaArtificial.	35
Tabla 7: Tarjeta CRC para la clase Grafo.	36
Tabla 8: Tarea de ingeniería # 1.	44
Tabla 9: Caso de prueba para el camino básico # 1.	48
Tabla 10: Caso de prueba para el camino básico # 2.	48
Tabla 11: Caso de prueba para el camino básico # 3.	49
Tabla 12: Caso de prueba para el camino básico #4.	50
Tabla 13: Caso de prueba para el camino básico #5.	50
Tabla 14: Caso de prueba de aceptación seleccionar características relevantes.	52
Tabla 15: Resultados de la estratificación realizada con la selección de características relevantes.	55
Tabla 16: Tabla comparativa de los resultados de los procesos de estratificación realizados (por estratos).	56

INTRODUCCIÓN

El progresivo avance tecnológico ha influido en el incremento de las bases de datos; este incremento, cuantitativo y cualitativo, es posible debido al progreso en las tecnologías para la adquisición y almacenamiento de los datos. Por consiguiente, estos beneficios han superado significativamente la capacidad de analizar, procesar y obtener los conocimientos de estos. Propiciando el incremento del desarrollo en diferentes campos del saber, la medicina, la agricultura (1), la meteorología (2), el turismo (3), la salud pública (4) (5), las geociencias, por citar algunas.

Muchos son los dominios donde se almacenan grandes volúmenes de información en bases de datos centralizadas y distribuidas, como, por ejemplo: archivos de imágenes, centros de investigación, bioinformática, cuidados médicos, finanzas e inversión, fábricas. La mayor parte de la de la información que se maneja en cualquier tipo de disciplina tiene una componente geoespacial. Se trata de información a la cual puede asignarse una posición geográfica, y por tanto viene acompañada de información adicional relativa a su localización. Por lo que es necesario la utilización de disímiles herramientas informáticas; una de las herramientas de gran impacto en los últimos años es los SIG. (6)

Los SIG dentro del panorama relativo a la geografía han sido vital para impulsar el desarrollo tecnológico de la sociedad y hacerla llegar hasta su lugar actual. Son capaces de ubicar un objeto determinado en el espacio; encontrar dónde 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. (7)

Los SIG en el ámbito de la salud poseen gran utilidad, debido a que 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 (8). Estos sistemas permiten identificar la posición geográfica de asentamientos de salud y de grupos poblacionales que poseen un alto riesgo de enfermar o morir en edades tempranas. Para realizar un análisis de las situaciones de salud existentes, es necesario conocer con el mayor grado de exactitud posible las características asociadas a cada territorio, partiendo de indicadores ambientales, demográficos, socioeconómicos entre otros de impacto que constituyen la base en el establecimiento de la estratificación territorial.

La estratificación territorial es considerada 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. Además, es una metodología 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) (9).

Cuba no está exenta de las transformaciones substanciales que suscitan en la contemporaneidad los avances en las TIC². El proceso de estratificación de territorios se desarrolla mediante soluciones informáticas por separado, primero se realiza el análisis estadístico y luego se presentan los resultados en mapas temáticos utilizando los SIG (10), lo que reduce la eficiencia del trabajo.

En aras de contribuir con el proceso de estratificación de territorios en la UCI³, en la Facultad 3 se desarrolló un componente para la estratificación de territorios utilizando los SIG que contribuye al mejoramiento de la capacidad de gestión de las entidades de salud. Este componente arrojó como resultado un sistema que permite integrar datos de variada naturaleza para el análisis y construcción de estratos, la solución del mismo 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.

Sin embargo, para realizar un estudio estratificado, los atributos son seleccionados por el investigador de forma manual, sin tener en cuenta su naturaleza, lo que puede introducir sesgo en la estratificación o datos redundantes e irrelevantes y por tanto no caracterizar bien al territorio. Cuando se incorporan todos los atributos el costo computacional es elevado, incidiendo en el tiempo en que se obtiene la información.

La selección de atributos relevantes permite la generación automatizada de los subconjuntos de rasgos principales que describen los datos. Consiste en encontrar el subconjunto de atributos del conjunto de

2 Siglas para el término: Tecnologías de la Información y las Comunicaciones.

3 Siglas para el término: Universidad de las Ciencias Informáticas.

datos original que mejor describe los objetos del dominio; tiene como fin reducir la dimensionalidad del conjunto de rasgos a través de la selección del subconjunto de rasgos de mejor desempeño bajo algún criterio de clasificación (11). Por tanto, la gran cantidad de información tanto irrelevante como redundante no brinda la información precisa con respecto a un conjunto determinado de características.

A partir del diagnóstico preliminar de la situación actual, se identifica el siguiente **problema a resolver**: el tratamiento a las características de los territorios en el proceso de estratificación sin tener en cuenta su relevancia, limita la obtención de información precisa.

A los efectos de la investigación, la información precisa es la actividad que satisface las necesidades de conocimientos mediante el análisis de datos. Para ello se utilizan indicadores de información precisa, que están basados en el tiempo, que sería el tiempo de respuesta utilizando la selección de atributos relevantes y sin utilizarlos, otro indicador sería la calidad, la cual está enfocada en el conjunto de características que caracterizan bien a los territorios cuando se construyan los conjuntos homogéneos (dentro del grupo) y heterogéneos (fuera del grupo).

Se delimitó como **objeto de estudio**: el proceso de estratificación de territorios centrando su **campo de acción**: selección de atributos relevantes en el proceso de estratificación de territorios.

Para darle solución al problema planteado se trazó como **objetivo general**: desarrollar un componente para la selección de características relevantes sobre el proceso de estratificación de territorios que facilite la obtención de información precisa en estudios salubristas.

El objetivo general se desglosó en los siguientes **objetivos específicos**:

- Construir el marco teórico referencial de la investigación, relacionado con la selección de atributos relevantes.
- Diseñar e implementar algoritmos de selección de características relevantes para el proceso de estratificación de territorios.
- Implementar un componente de soporte a la selección de atributos relevantes.
- Verificar la solución informática propuesta aplicando diferentes pruebas (pruebas de caja blanca y pruebas de aceptación) y métricas.

Como **idea a defender** se plantea que, si se desarrolla un componente para la selección de características relevantes en el proceso de estratificación de territorios, entonces se facilitará la obtención de información precisa en estudios salubristas.

En la consecución de los objetivos trazados se utilizaron los siguientes **métodos**:

- **Métodos generales:** El método hipotético-deductivo para elaborar la hipótesis de investigación y proponer líneas de trabajo a partir de resultados parciales; el método sistémico para el desarrollo del sistema computacional y lograr que los elementos que formen parte de la aplicación real sean un todo que funcione de manera armónica; el método histórico lógico permitió realizar un estudio de las principales herramientas que realizan procesos de estratificación de territorios y el dialéctico para el estudio crítico de los trabajos anteriores.
- **Métodos lógicos:** El método analítico-sintético al descomponer el problema de investigación en elementos por separado y profundizar en el estudio de cada uno de ellos, para luego sintetizarlos en la solución de la propuesta; el método inducción -deducción como vía de la constatación teórica durante el desarrollo de la tesis; el método de modelado para el desarrollo de los algoritmos.
- **Métodos empíricos:** El método experimental para comprobar la utilidad de los resultados obtenidos.

Posibles resultados: Obtener un componente para la selección de características relevantes en el proceso de estratificación de territorios que facilite la obtención de información precisa en estudios salubristas.

Validación

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

El presente trabajo de diploma se verá estructurado de la siguiente manera: introducción, tres capítulos que constituyen el cuerpo fundamental de la tesis, conclusiones, recomendaciones, bibliografía y anexos. Para tener un conocimiento de manera general de lo que se abordará se describe a continuación una síntesis de cada capítulo.

Capítulo 1: Referentes teóricos sobre la estratificación de territorios.

En este capítulo se analizan los principales conceptos relacionados con el objeto de estudio, para lograr un mejor entendimiento del trabajo a desarrollar. Se hará referencia a la selección de atributos relevantes, la definición de qué es un problema de clasificación y qué es dentro de un problema de clasificación las características de atributos relevantes. Además, se realiza un estudio de la metodología, herramientas, tecnologías y lenguajes a utilizar en el desarrollo de la solución.

Capítulo 2: Componente para la selección de características relevantes sobre el sistema de estratificación de territorios.

En este capítulo se describe la propuesta de solución, se especifican requisitos funcionales y no funcionales que serán un elemento a tener en cuenta para la implementación de la misma, y se detallan aspectos relacionados con su diseño y arquitectura. Se especifican los patrones de diseño aplicados y los artefactos derivados de la metodología de desarrollo de software seleccionada.

Capítulo 3: Verificación de la solución.

Este capítulo se centra en los procesos de implementación y prueba de las funcionalidades del producto a desarrollar. Contiene los casos de prueba y otras técnicas de prueba utilizadas para validar las funcionalidades desarrolladas 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 exponen los principales conceptos que sirven de base para adentrarse en el tema de la estratificación de territorios. Se expondrán varias definiciones, que permitirán saber que son los SIG, el análisis espacial y técnicas que existen. Se conocerá qué es un problema de clasificación y qué es dentro de un problema de clasificación la selección de atributos relevantes. Se hará referencia a la selección de atributos relevantes y se mencionarán algunos de los diferentes enfoques y técnicas para realizar esta selección. Además de una breve reseña de lo que es un grafo. Por último, se realiza un estudio de la metodología, herramientas, tecnologías y lenguajes a utilizar en el desarrollo de la solución.

Estratificación

En la estratificación se integran varios tipos de información: demográfica, social, económica, de eventos de salud y condiciones asociadas y otros en los que se incluyen datos sobre el sistema de salud y la opinión de la población sobre su salud y los servicios que recibe. Para lograr una mayor racionalidad en el análisis de las diferentes situaciones de salud, es necesario conocer con el mayor detalle posible, las características de cada una de las unidades territoriales, así como de sus diferentes grupos poblacionales, tanto en sus variables demográficas, como socio-económicas. Todos estos elementos son los que constituyen la caracterización de un territorio, que sirve de base para el establecimiento de la estratificación territorial.

La estratificación territorial es un conjunto de analogías que dan lugar a subconjuntos de unidades agregadas, denominadas estratos. Un estrato 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. En cada estrato existe una igualdad interna con diferencias o desigualdades externas.

Posibilita la separación de los elementos de un conjunto en niveles o estratos representativos de un universo, con desigualdades, dejando bien definidos los criterios de diferenciación cuantitativa y cualitativa. Constituye un nivel de agregación que permite controlar la heterogeneidad de los territorios y perfeccionar el conocimiento y la interpretación de la información generada, con una visión "desagregada" de esta al nivel territorial más general, o con una visión agregada al nivel territorial más particular (nivel local).

La estratificación territorial es una metodología que consiente en dimensionar espacialmente los eventos a través de un proceso de agregación y desagregación de los territorios a valorar, partiendo de variables

seleccionadas para dichos territorios que permitan agregaciones (por homologías de las características) o desagregaciones (por heterogeneidades de estas). (9)

En esencia la estratificación es una poderosa estrategia de búsqueda que facilita entender cómo influyen los diversos factores o variantes que intervienen en una situación problemática, de forma que puedan localizar diferencias, prioridades y pistas que permitan profundizar en la búsqueda de las verdaderas causas de un problema.

Se ha estimado que cerca del 90% de la información de quienes toman decisiones y definen políticas en los gobiernos locales, está relacionado con una ubicación geográfica, o sea que está íntimamente asociado a las variables espaciales. Existen varias herramientas de análisis que posibilitan resolver problemas asociados a la distribución espacial, como es el caso de la utilización de los SIG en el análisis espacial de enfermedades. (12).

1.1. Sistemas de Información Geográfica

Los SIG pueden definirse de forma provisional como sistemas que almacenan datos espaciales para su consulta, manipulación y representación. (13) La razón fundamental para utilizar un SIG es la gestión de información espacial. El sistema permite separar la información en diferentes capas temáticas y las almacena de forma independiente, permitiendo trabajar con ellas de manera rápida y sencilla, y facilitando al profesional la posibilidad de relacionar la información existente a través de la topología de los objetos, con el fin de generar otra nueva que no se pueda obtener de otra forma.

Estas herramientas permiten no sólo trabajar con grandes volúmenes de datos, sino que además permiten almacenarlos física y completamente, y en varios niveles, por ejemplo, público e institucional. Los datos son almacenados y representados de manera independiente. También permiten trabajar e integrar información de diferentes fuentes y escalas, y datos tanto espaciales como no-espaciales. Además, todos estos datos pueden ser analizados a la vez, incluso repetidamente, y de una forma rápida, racional y fácilmente inteligible para el usuario, permitiendo así una evaluación ágil y sencilla. No sólo pueden integrarse conjuntos de datos diferentes, sino que también pueden ser integrados varios procedimientos distintos. Esto no sería posible mediante ningún método manual. Además, evita actividades redundantes.

Dentro del amplio aspecto de características pueden detectarse 3 puntos convergentes que afirman que el SIG:

- Es un sistema de información, compuesto por hardware, software, datos, procedimientos y recursos humanos, destinado a soportar los procesos de toma de decisiones.
- Trabaja con una base de datos espaciales alfanuméricos y cartográficos.
- Cuenta con funciones especializadas de captura, almacenamiento, transformación, modelización, análisis y presentación de datos espaciales, para la resolución de problemas de naturaleza geográfica.

Los SIG permiten hacer un análisis exhaustivo del territorio en los ámbitos más diversos. Son herramientas versátiles, con un amplio campo de aplicación en cualquier actividad que conlleve un componente espacial.

1.1.1. Conceptos básicos asociados a los SIG

La **cartografía** es el arte de trazar mapas geográficos; ciencia que los estudia y se encarga de la elaboración de estos. (14).

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 (14).

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 (15).

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. (15)

1.1.2. Panorama actual de aplicaciones SIG

Actualmente existe una gran diversidad de SIG, cada uno de ellos con numerosas alternativas, por lo que resulta complejo elegir una que se ajuste a cada necesidad. Para esto es necesario tener una visión global 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 (6) 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

GRASS⁴ es el proyecto SIG libre más antiguo, 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 (QGIS) es la más destacable, constituyendo una interfaz de usuario sencilla para GRASS. (16)

Software SAGA

SAGA⁵ es un software SIG de escritorio, multiplataforma, desarrollado en Alemania y con un fuerte enfoque hacia el análisis de datos geo-espaciales. 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 (17).

Software Qgis

QGIS es una aplicación SIG con grandes potencialidades para la edición de mapas, multiplataforma y desarrollado utilizando Qt Toolkit y C++. Ofrece muchas características SIG, entre las que se encuentran (18):

- Permite crear, editar, administrar y exportar mapas vectoriales en varios formatos.
- Permite realizar análisis de datos espaciales de PostgreSQL/PostGIS usando el complemento de *Python fTools*.

4 Siglas en inglés para el término: Geographic Resources Analysis Support System.

5 Siglas en inglés para el término: System for Automated Geoscientific Analyses.

- 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.
- Permite la integración de componentes 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, por lo que facilita la representación en mapas temáticos como resultado de los procesos asociados a la georreferenciación y análisis.

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 e integración de la solución.

1.2. 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 tuvo en cuenta para el análisis, los algoritmos de agrupamiento que estas emplean, la forma de representación de los resultados y los tipos de indicadores que analizan.

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

Representación gráfica de los resultados	Mapas temáticos	Burbujas, Centroides, Mapas temáticos	Mapas temáticos
Tipo de indicadores que utiliza	Estadístico	Estadístico	Estadístico
Fuente de información	Obtiene la información a través de una hoja de cálculo	Presenta un conjunto estático de datos	Presenta un conjunto estático de datos
Tipo de licencia	Privativo	Libre	Libre

Luego de llevar a cabo una comparación entre las herramientas existentes que realizan procesos de estratificación se concluye que presentan deficiencias para incorporar la naturaleza espacial de los datos, en la obtención de indicadores de diferentes fuentes de datos, y no tienen en cuenta la relevancia de atributos, 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 complemento para el software SIG seleccionado, que permita tener en cuenta la selección de características relevantes.

1.3. Atributos relevantes

La selección de atributos relevantes puede ser vista como uno de los problemas más importantes en el campo del aprendizaje automático y no supervisado. Dado que la propuesta de solución de esta investigación va enfocada a los resultados que se pretenden obtener con la aplicación de la selección de estos, es imprescindible abordar los elementos fundamentales del mismo. La selección de atributos es un campo de investigación y desarrollo productivo desde los años setenta, donde confluyen distintas áreas como el reconocimiento de patrones (19) (20) (21), el aprendizaje automático (22) (23) (24) y la minería de datos (24).

La selección de rasgos consiste en encontrar el subconjunto de atributos del conjunto de datos original que mejor describe los objetos del dominio; tiene como meta reducir la dimensionalidad del conjunto de rasgos a través de la selección del subconjunto de rasgos de mejor desempeño bajo algún criterio de clasificación (11). Las técnicas de selección de características se aplican en entornos diferentes, por ejemplo la clasificación de textos, recuperación de imagen (25). El proceso de selección de atributos, además de preceder a la clasificación, suele estar presente en las etapas previas de la minería de datos, ya sean supervisadas o no (26), como la regresión, el agrupamiento y las reglas de asociación (27). Este proceso se desarrolla eliminando rasgos irrelevantes y redundantes (28) (29), que facilita una mejor representación de la información original, reduce el costo computacional y mejora la generalización del algoritmo de aprendizaje (30).

1.3.1. Enfoques y técnicas para la selección de atributos relevantes

Los procedimientos de selección de rasgos constan de dos componentes principales: la función de evaluación y el método de generación de subconjuntos (basado en un proceso de búsqueda). Existen diferentes enfoques y técnicas para seleccionar atributos relevantes; en el presente trabajo como método de generación se utilizará la metaheurística basada en ACO.

Algoritmos de ACO

Los algoritmos de ACO son una metaheurísticas poblacional que puede ser usada para encontrar soluciones aproximadas a problemas complejos de optimización discreta (31). Durante cada ciclo, un número de hormigas artificiales construyen secuencialmente soluciones de forma combinada, aleatoria y golosa. Cada hormiga selecciona el próximo elemento a ser incorporado en su solución parcial sobre la base de una heurística y la cantidad de feromona asociada con este elemento. La heurística provee el valor de una solución candidata específica. La feromona representa la memoria del sistema, y está relacionada con la presencia de ese elemento en soluciones previamente construidas (de esta forma la intensidad del rastro de feromona está relacionado con cuántas hormigas habían decidido anteriormente seguir ese camino). La aleatoriedad (el hecho de hacer algún tipo de selección al azar) es usada para facilitar la construcción de una variedad de soluciones diferentes. Se define una distribución de probabilidad sobre todos los elementos que pueden ser incorporados a la solución parcial actual favoreciendo los mejores elementos. En particular, un elemento con una buena evaluación heurística y un alto nivel de feromona es más propenso a ser seleccionado (30).

Cada vez que una hormiga selecciona un elemento actualiza el nivel de feromona de éste, primero substrayendo una fracción de su valor, imitando la evaporación de la feromona y luego adicionando un nuevo valor. Cuando todas las hormigas han construido una solución completa, el procedimiento se reinicia manteniendo los valores del nivel de feromona actualizado. Esto es repetido para un número preestablecido de ciclos o hasta algún otro criterio de parada.

Los algoritmos de ACO reproducen el comportamiento de las hormigas reales en una colonia artificial. Estos han sido aplicados a un gran número de problemas cuyas soluciones no son polinómicas, como el vendedor ambulante, problemas de ruteo en redes de telecomunicaciones, planificación de tareas, etcétera. Los algoritmos ACO son procesos iterativos, en cada iteración se "lanza" una colonia de hormigas y cada una de ellas construye una solución al problema.

Las hormigas construyen las soluciones de manera probabilística, guiándose por un rastro de feromona artificial y por una información calculada a priori de manera heurística (32). Jensen y Shen plantean que el uso de estas técnicas para el cálculo de reductos debido a que las hormigas pueden descubrir las mejores combinaciones de atributos en la medida en que atraviesan el grafo (33). Computacionalmente este modelo basado en colonia de hormigas se modela como un grafo.

1.3.2. Representación mediante un grafo del funcionamiento de las hormigas

Grafos

La teoría de grafos, se ha convertido en una herramienta matemática indispensable en campos tan diversos como la investigación operativa, la lingüística, la química, la física, la genética, la teoría de redes o la teoría de la decisión. Es por ello que, para casi cualquier rama de la ciencia, se hace indispensable el conocimiento, a través de conceptos globales, de las ideas básicas que sustentan a la teoría de grafos. Las gramáticas de grafo pueden ser aplicadas en la solución de varios problemas, en particular en los SIG, se puede afirmar que la teoría de grafos está estrechamente vinculada con los Sistemas de Información Geográfica (34).

Definición de grafo

- Se denomina grafo G al par $G = (V, E)$, en el que V es un conjunto no vacío de elementos denominados vértices (también llamados nodos o puntos) y E es el conjunto finito de pares no ordenados de elementos de V denominados aristas (igualmente líneas) (34).

- Un vértice v_j es adyacente (vecino) a otro vértice v_i si $\exists (v_i, v_j) \in E$. Además, se dice que la arista (v_i, v_j) incide en el vértice v_j .
- Se denomina grado de un vértice v de G al número de aristas que inciden en v . A un vértice de grado 0 se le denomina vértice aislado. A un vértice de grado 1 se le denomina vértice terminal o extremo.
- Sean $G = (V, E)$ y $G' = (V', E')$ dos grafos, si $V' \subseteq V, E' \subseteq E$ se dice que G' es un subgrafo de G y G es un supergrafo de G' (35).

A continuación, se exponen las tipologías más comunes de los grafos.

Grafo Simple

Se denomina grafo simple al grafo $G = (V, E)$ que verifica que para todo $u, v \in V(G)$, existe a lo sumo una única arista $\langle u, v \rangle$ de E que los une (34).

Grafo Orientado

Un grafo orientado D , también denominados redes o digrafos, se define como un par $(V(D), A(D))$, donde $V(D)$ es un conjunto finito no vacío de elementos llamados vértices, y $A(D)$ es una familia finita de pares ordenados de elementos de $V(D)$ llamados arcos (o aristas orientadas o di-aristas) (34).

Grafo Regular

Se llama grafo regular a un grafo cuyos vértices tienen todo el mismo grado. Si el grado de cada vértice es r , se tiene un grafo regular de grado r (34).

Grafo Bipartito

Sea un grafo G y sea su conjunto de vértices $V(G)$ que puede ser expresado como la unión disjunta de dos subconjuntos de vértices V_1 y V_2 de forma que cada arista de G une un vértice de V_1 con otro de V_2 , entonces se dice que G es un grafo bipartito y se escribe $G(V_1, V_2)$ (34).

Grafo Completo

Un grafo completo es un grafo simple en el que cualquier par de vértices son adyacentes (34).

1.3.3. Problema de clasificación

Según la literatura especializada de epidemiología se denomina conglomerado o *clúster* a un exceso de casos de enfermos diagnosticados superior a lo esperado en un área geográfica determinada

(conglomerado espacial), en un período de tiempo limitado (conglomerado temporal), o considerando ambos dominios (conglomerado espacio-temporal) (36).

En (37) (38) se presenta una introducción a los algoritmos de agrupamiento, además se puede consultar en (39) (40) el enfoque estadístico de las técnicas de clasificación no supervisada.

La meta de todo algoritmo de agrupamiento consiste en asignar cada objeto a un sistema finito de subconjuntos o *clústeres* que usualmente no se interceptan entre sí y cuya unión es igual al conjunto de objetos con la posible excepción de *outliers*, de modo tal que objetos similares pertenezcan al mismo *clúster*, mientras que los objetos de *clústeres* diferentes sean lo menos semejantes posible.

Los algoritmos de agrupamiento han sido empleados en reconocimiento del habla, en segmentación de imágenes y visión por computador (41) (42); en minería de datos para extraer conocimiento desde fuentes de datos, en la recuperación de información y minería de textos (43) (44) (45) (46), en minería de datos geoespaciales (47) (48) (49) y (48), en análisis de datos heterogéneos, en aplicaciones Web (50) (51) (52), en biología computacional para el análisis de ADN1 (53), por solo mencionar algunas.

Los algoritmos de agrupamiento pueden dividirse en varias categorías según el procedimiento que utilizan para agrupar los objetos (54):

- Algoritmos jerárquicos, que pueden ser aglomerativos y divisivos.
- Métodos por partición, entre ellos: algoritmos de reubicación, agrupamientos probabilísticos, métodos de k-medoides y métodos k-Medias (k-Means).
- Algoritmos basados en densidad, entre ellos los algoritmos de agrupamiento por conectividad basados en densidad y los agrupamientos basados en funciones de densidad.
- Métodos basados en rejillas.

Existen diversas razones por las que los métodos no supervisados son de interés: convergen rápidamente y con buen desempeño en caso de que las características cambien poco en el tiempo, permiten categorizar elementos, son útiles cuando el etiquetado de un conjunto grande de muestras no es factible y contiene datos de bajo grado de relevancia para la solución de un problema, permitiendo la selección de atributos relevantes determinando tendencias o patrones de comportamiento.

1.4. 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.

1.4.1. Lenguaje de modelado

UML⁷ 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. (55). 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.

1.4.2. Herramienta CASE

Las herramientas CASE⁸ 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 para el desarrollo de un software (56).

Visual Paradigm

Es una herramienta de diseño que facilita el desarrollo de software. Ofrece un paquete completo útil para la captura de requisitos, la planificación del software, la planificación de pruebas, el modelado de clases y el modelado de datos (57).

Las principales características de la herramienta son:

- ❖ Soporta las últimas versiones del UML.
- ❖ Posee un poderoso generador de documentación y reportes en formato PDF, HTTP y JPG.

⁷Siglas para el término: Lenguaje Unificado de Modelado.

⁸ Siglas en inglés para el término: Computer Aided Software Engineering.

- ❖ 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 8.1 como herramienta para el modelado UML. Posibilita la realización de los diagramas necesarios para el desarrollo y mejor entendimiento de la aplicación. Al ser seleccionado el lenguaje de modelado UML, es conveniente tener en cuenta su vinculación con Visual Paradigm, resaltando que este último presenta abundante documentación y demostraciones interactivas.

1.4.3. Lenguaje de programación

Se seleccionó el lenguaje de programación *Python* a partir de la posibilidad que brinda QGis para integrar componentes implementados en este lenguaje.

Python

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.

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. La decisión de seleccionar como IDE, Pycharm en su versión 3.4, está dada a que ofrece autocompletación inteligente de código, comprobación de errores sobre la marcha, soluciones rápidas y fácil navegación en el proyecto. 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.

1.4.4. Gestor de base de datos.

Los GBD⁹ 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 bases de datos. Estos softwares facilitan el proceso de definir, construir y manipular bases de datos para diversas aplicaciones.

⁹Siglas para el término: Gestores de Bases de Datos.

PostgreSQL

PostgreSQL es un sistema de GBD objeto-relacional, de propósito general, multiusuario y de código abierto, que soporta gran parte del estándar SQL¹⁰ 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. 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. PostGIS incluye un conjunto de operaciones para realizar consultas espaciales muy bien optimizadas por sus índices R-Tree y su integración con el planificador de consultas de PostgreSQL. Utiliza las librerías Proj4 para dar soporte a la transformación dinámica de coordenadas y la biblioteca GEOS 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.

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 para la parte del servidor y un agente para lanzar scripts programados. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas Unix), y puede encriptarse mediante SSL para mayor seguridad [66].

¹⁰ Siglas en inglés para el término: Structured Query Language.

1.4.5. Metodología de desarrollo

Se hace necesario contar con proceso bien detallado para la realización de un software y se necesita aplicar una metodología capaz de realizar el control total de los productos. Existen una gran variedad de metodologías debido a que estas pretenden guiar a los desarrolladores, sin embargo los requisitos de un software son muy variados y cambiantes (58) .

Las metodologías se dividen en dos grupos: las metodologías tradicionales y las ágiles. En caso de las tradicionales se centran en la definición detallada de las tareas y procesos que se deben realizar, herramientas a utilizar, y necesitan una extensa documentación, pretendiendo prever todo con anterioridad, también dependen de un equipo de desarrollo amplio. En el caso de las ágiles es más importante lograr que un producto de software se desarrolle con la calidad requerida, en lugar de realizar una buena documentación. En este tipo de metodología se cuenta con un equipo de desarrollo pequeño y el cliente está presente en todo momento y colabora con el proyecto (58).

Programación Extrema

XP¹¹ es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores y propiciando un buen clima de trabajo. Además se basa en retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico (59).

Características de la metodología XP (60):

- XP es una metodología “liviana” que no tiene en cuenta la utilización de 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.

¹¹Siglas para el término: Programación extrema.

- XP define HU¹² 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 CRC¹³.
- **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.

¹²Siglas para el término: Historia de usuarios.

¹³ Siglas para el término: Clase-Responsabilidad- Colaboración.

Partiendo de lo antes expuesto, se pudo apreciar que XP responde a las necesidades principales de tiempo, entorno y cantidad de 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.

1.5. Conclusiones parciales

Luego de realizar un estudio de los principales conceptos que sirven de base para adentrarse en el tema de la estratificación de territorios, se concluye lo siguiente:

- Mediante el desarrollo de este capítulo se obtuvo un mejor dimensionamiento del problema partiendo del análisis de los conceptos principales asociados a la solución.
- La revisión del panorama actual de los softwares para SIG condujo a la selección de Qgis, debido a que posee funcionalidades que pueden ser reutilizadas e integradas a la solución.
- El estudio de variados sistemas que soportan procesos de estratificación de territorios permitió evidenciar la necesidad de desarrollar un complemento que tenga en cuenta la selección de características relevantes.
- El uso de los conceptos relacionados con la teoría de grafo contribuye a comprender de forma más exacta el comportamiento de la metaheurística ACO.
- La metodología XP se escogió para guiar todo el proceso de desarrollo de la solución, permitiendo tener una retroalimentación entre el cliente y el equipo de desarrollo.

CAPÍTULO 2. COMPONENTE PARA LA SELECCIÓN DE CARACTERÍSTICAS RELEVANTES SOBRE EL SISTEMA DE ESTRATIFICACIÓN DE TERRITORIOS

En este capítulo se muestran los elementos concernientes con el desarrollo de un componente informático para la selección de características relevantes sobre el sistema de estratificación de territorios y se detalla 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. Se define la arquitectura y los principales patrones de diseño utilizados en el desarrollo de la solución.

En la Figura 1 se presenta el modelo conceptual de la propuesta de solución que se propone.

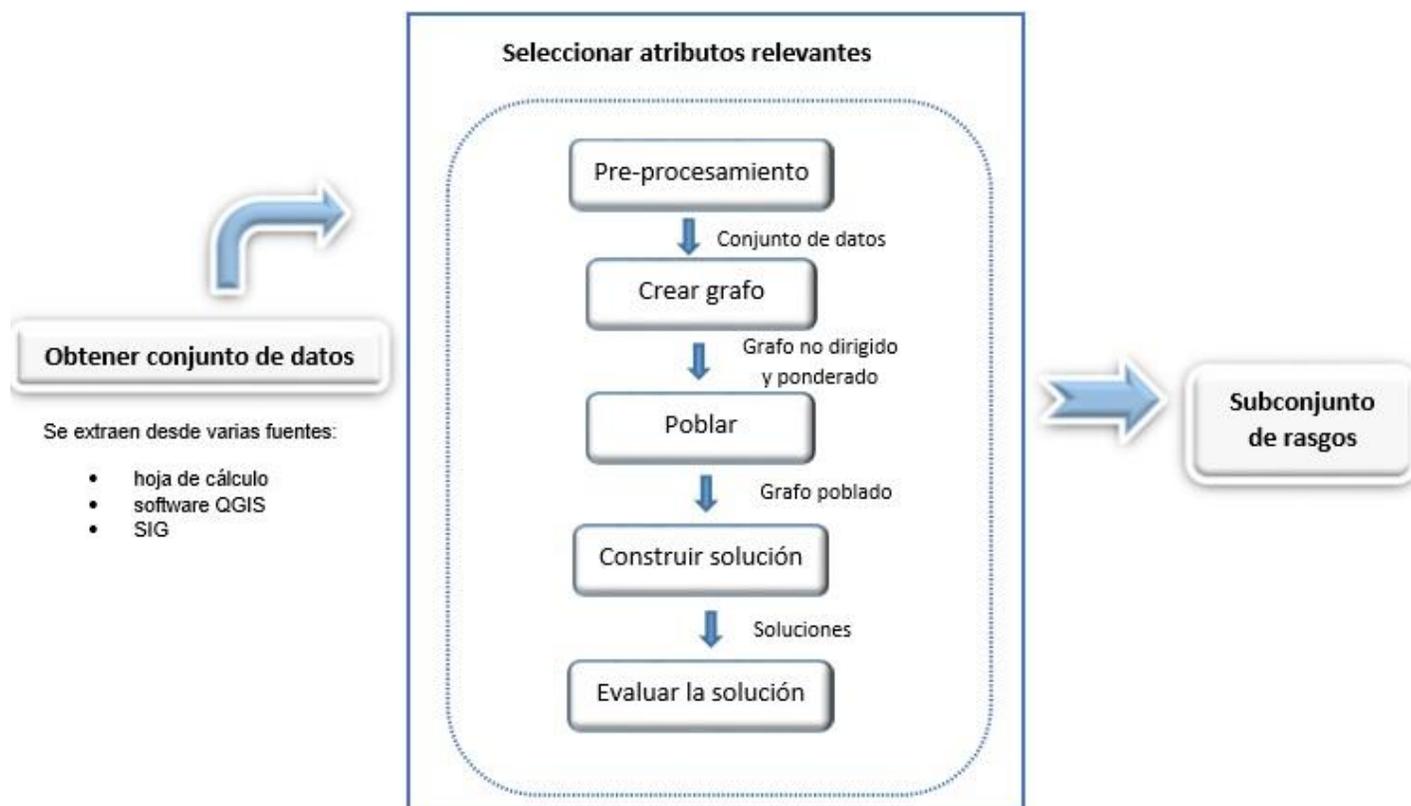


Figura 1: Modelo conceptual de la propuesta de solución.

A continuación se detalla cada componente del modelo, así como las relaciones entre ellos.

El modelo propuesto recibe como entrada un conjunto de datos, de disímiles fuentes: hojas de cálculos, software Qgis y SIG. Estos datos estarán conformados por los objetos y los rasgos de cada uno de ellos.

Luego se procede a realizar el proceso de selección de características relevantes siguiendo la metodología de la metaheurística ACO. Computacionalmente este modelo, se representa como un grafo completo.

Para realizar la evaluación de la solución propuesta de cada subconjunto de la hormiga en esta investigación se pretende utilizar las características más visitadas.

Finalmente se obtiene como resultado el subconjunto de características que mejor describen a los objetos.

2.1. Selección de atributos relevantes

En la presente investigación el proceso de selección de características relevantes se desglosa en los siguientes pasos:

- **Preprocesamiento:** se obtiene el conjunto de características y el espacio.
- **Crear grafo:** se crea un grafo completo.
- **Poblar grafo:** se crean las hormigas y se ubican en el grafo.
- **Construir soluciones:** se construye la solución.
- **Evaluar solución:** se selecciona el conjunto de características que mejor describa el problema.

Preprocesamiento de los datos

Se obtiene un conjunto de características y el espacio.

Para determinar la similitud entre los rasgos se construye una matriz de similitud a partir de la ecuación 1:

$$sim(A, B) = \left| \frac{\sum_{i=1}^p a_i b_i}{\left(\sqrt{\sum_{i=1}^p a_i^2}\right)\left(\sqrt{\sum_{i=1}^p b_i^2}\right)} \right| \quad (1)$$

Donde A y B son los rasgos, p la dimensionalidad, a_i y b_i son los valores que toman los rasgos A y B respectivamente para cada objeto de la matriz de datos (61).

Para el cálculo de la similitud se diseñó el algoritmo 1.

Algoritmo 1 similitudMatriz ()

Entrada: Matriz de datos (matriz)

Salida: Matriz de similitud

1. similitud = crearMatriz() se debe crear una matriz cuadrada
 2. Para todo x, y \in matriz:
 3. $V = \text{similitud}(x, y)$ //en esta se utiliza la ecuación 1
 4. $\text{similitud}[x, y] = v$
 5. $\text{similitud}[y, x] = v$
 6. **Fin** para
 7. **Fin**
 8. **Retornar** similitud
-

Crear grafo

A partir de las características que describen a los objetos se crea un grafo completo. Un grafo es un par (V, E) , donde V es el conjunto de vértices, que van hacer los rasgos y E el conjunto de aristas, al ser un grafo completo van a establecer aristas entre cada par de vértice. Las aristas serán arcos ponderados, el costo asociado a esta arista es la similitud que existe entre estos rasgos a partir de la ecuación 1.

El grafo se construye a partir del algoritmo 2

Algoritmo 2 construirGrafo()

Entrada: Matriz de datos (matriz)

Salida: grafo no dirigido y ponderado (G)

1. $G = (V = \{\}, E = \{\})$
 2. $L = \text{extraerRasgos}(\text{matriz})$
 3. Para todo $x \in L$:
 4. $V.\text{adicionar}(x)$
 5. **Fin** Para
 6. $E = \text{similitudMatriz}()$
 7. **Fin**
 8. **Retornar** G
-

Poblar

Proceso donde se crean las hormigas y se ubican en el grafo. Para realizar la creación de estas en varias bibliografías consultadas proponen las siguientes reglas para determinar el número de hormigas (k) como una función de la cantidad de rasgos (nf), $k=f(nf)$ (32).

En esta investigación se utiliza la clasificación del problema de selección de rasgos (62) donde se clasifican en tres categorías: pequeña escala si $nf \in [0, 19]$, mediana escala si $nf \in [20, 49]$, y escala grande si $nf \geq 50$.

Se diseñó el algoritmo **3 y 4** calcular la cantidad de hormigas y para crear las hormigas.

Algoritmo 3 cantidadHormigas()

Entrada: cantidad de rasgos (nf)**Salida:** cantidad de hormigas (k)

1. Si $nf \in [0, 19]$:
2. $k \leftarrow nf$
3. Si $nf \in [20, 49]$:
4. Si $0.66 * nf \leq 24$:
5. $k \leftarrow 24$
6. Sino $k \leftarrow \text{Round}(0,66 * nf)$ // Donde *Round* (x) denota el entero más cercano a x .
7. Sino:
8. Si $0,5 * nf \leq 33$:
9. $k \leftarrow 33$
10. Sino $k \leftarrow \text{Round}(0,5 * nf)$
11. **Fin**
12. **Retornar** k

Una vez creadas las hormigas se deben distribuir en el grafo, para ello definió el algoritmo **5**

Algoritmo 4 distribuirHormigas()

Entrada: cantidad de hormigas (k), Grafo (G)**Salida:** G

1. Si $k = \text{len}(G.V())$:
2. distribuirHormigasPeq(k) // se ubica una hormiga en cada nodo
3. Si $k < \text{len}(G.V())$:
4. distribuirHormigasAleatoria(k) //se ubican las hormigas en nodos aleatorios

5. Si $k > \text{len}(G.V())$:
6. distribuirHormigasPeq(k)
7. distribuirHormigasAleatoria($\text{len}(G.V())-k$)
8. **Fin**
9. **Retornar G**

Construir solución

Luego de la distribución cada hormiga debe construir su solución, a medida que recorre el grafo.

En el algoritmo 5 se puede evidenciar el procedimiento antes mencionado.

Algoritmo 5 *run()*

Entrada: Grafo (G), nodo inicial (nodo)

Salida: soluciones (rasgos)

1. Mientras $\text{fin}=\text{false}$:
2. candidato= reglaDeTransicion(nodo)
3. rasgos.add(candidato)
4. nodo= candidato
5. visitado(candidato)
6. Si $\text{reglaDeTransicion}(\text{nodo})=\emptyset$
7. $\text{fin}=\text{false}$
8. **Fin**
9. **Retornar** rasgos

En cada iteración se actualiza la feromona (p) mediante:

$$\tau_i(t+1) = (1-p)\tau_i(t) + \frac{FC_{[i]}}{\sum_{j=1}^n FC_{[j]}} \quad (2)$$

Donde n es el número original de características, $\tau_i(t)$ y $\tau_i(t+1)$ son las cantidades de valores de feromonas de la característica i , p es un parámetro de evaporización de feromonas y $FC_{[i]}$ es el contador correspondiente a la característica i (61).

Evaluar solución

Una que vez todas las hormigas finalicen su recorrido se debe seleccionar el subconjunto que mejor describa el problema. En esta investigación se determinó que este estará formado por los atributos que fueron más visitados, o sea los nodos más visitados.

Luego de realizado el proceso de selección de atributos relevantes se muestra el conjunto de características seleccionadas para realizar el proceso de estratificación de territorios.

A partir del modelo propuesto y para garantizar aplicabilidad se desarrolla una solución informática, a continuación se describen los elementos, iniciando por los requisitos.

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” (63). La calidad con que se lleva a cabo la captura de los requisitos influye en todo el proceso de desarrollo del software repercutiendo en el resto de las fases de su desarrollo. Además, ayuda a tomar mejores decisiones de diseño y arquitectura.

2.3. Requisitos funcionales

Los RF¹⁴ identificados para el proceso de desarrollo son:

➤ **Requisitos funcionales**

RF1: Obtener el conjunto de características.

RF2: Seleccionar características relevantes.

RF2.1: Seleccionar algoritmo.

RF2.2: Seleccionar parámetros.

RF3: Visualizar características relevantes.

2.4. Requisitos no funcionales

Los RNF¹⁵ identificados son:

¹⁴Siglas para el término: Requisito funcional.

¹⁵Siglas para el término: Requisitos no funcionales.

➤ **Requisitos de Software**

RNF1: Se debe tener instalada la herramienta QGIS en su versión 2.6 o superior.

RNF2: Se debe tener instalado el GBD PostgreSQL en su versión 9.0 o superior.

RNF3: Se debe tener instalado el módulo PostGIS en su versión 2.1.5 o superior.

➤ **Requisitos de Hardware**

RNF4: La estación de trabajo debe contar con al menos 1,0 GB de Random Access Memory (RAM, por sus siglas en inglés).

RNF5: La capacidad mínima de espacio en disco debe ser 2.0 GB.

➤ **Requisitos de Usabilidad**

RNF6: 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**

RNF7: Debe tener una interfaz amigable y con apariencia profesional.

RNF8: La interfaz debe tener un diseño sencillo y ser de fácil comprensión para el usuario.

➤ **Restricciones de diseño e implementación**

RNF9: Se hace uso de la herramienta QGIS en su versión 2.6 e IDE Pycharm versión 3.4.

RNF10: El lenguaje de programación usado para la implementación es *Python*.

2.5. Fase de planificación

La metodología XP define como fase inicial la planificación. Durante esta etapa se lleva a cabo el proceso de identificación y confección de las historias de usuario, así como la familiarización del equipo de trabajo con las tecnologías y herramientas seleccionadas para el desarrollo del software. El cliente especifica la prioridad en que se deben implementar las historias de usuario, además de una estimación del esfuerzo. El resultado de la fase es un plan de entregas donde se realiza una estimación de las versiones que tendrá el producto en su realización, de manera tal que guíe su desarrollo.

2.5.1. Historias de usuarios

Las HU constituyen 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 específica o generales, añadirse nuevas o ser modificadas. Cada HU es comprensible y delimitada para que los programadores puedan implementarla en unas semanas (58)

Para hacer más comprensible las mismas, a continuación, se describe su leyenda:

- **Número:** Número de la historia de usuario incremental en el tiempo.
- **Nombre de la historia de usuario:** Nombre de la historia de usuario especificado por el programador.
- **Programadores responsables:** Personas involucradas en el desarrollo de las HU.
- **Prioridad en negocio (Baja, Media, Alta):**

Baja: Se le otorga a las HU que son de funcionalidades auxiliares y que son independientes del sistema.

Media: Se le otorga a las HU que son de funcionalidades a tener en cuenta, sin que estas tengan una afectación sobre el sistema que se esté desarrollando.

Alta: Se le otorga a las HU que son de funcionalidades fundamentales en el desarrollo del sistema.

- **Riesgo en desarrollo (Bajo, Medio, Alto):**
Bajo: Cuando en la implementación de las HU puedan existir errores, pero son tratados fácilmente y no afectan el desarrollo del sistema.

Medio: Cuando en la implementación de las HU puedan existir errores y retrasen la entrega del producto.

Alto: Cuando en la implementación de las HU pueda existir algún error y afecte la disponibilidad del sistema.

- **Puntos estimados:** Tiempo estimado que se demorará el desarrollo de la HU.
- **Iteración asignada:** Número de la iteración.
- **Descripción:** Breve descripción de la HU.
- **Observaciones:** Señalamiento o advertencia del sistema.

Luego de obtener las principales funcionalidades del sistema, se identificaron 3(tres) HU. En la tabla 2 se muestra una breve descripción de 1(una) de ellas.

Tabla 2: Historia de usuario. Seleccionar características relevantes.

Número:	2	Nombre de la historia de usuario:	Seleccionar características relevantes
Programadores Responsables:		Lisette Castro Palacio y Rosmery Hernández Peña	
Prioridad en negocio:	Alta	Puntos estimados:	2
Riesgo en desarrollo:	Alta	Iteración asignada:	1
Descripción:	<p>La aplicación debe ser capaz de analizar los datos y permitir seleccionar las características relevantes a partir de:</p> <ul style="list-style-type: none"> • La construcción del grafo a partir de los rasgos que se presentan. • La asignación de las hormigas a cada uno de los nodos. • La construcción de la solución. • La obtención de las características relevantes, en función de la cantidad de visitas asociadas a cada nodo. 		
Observaciones:			

2.5.2. Estimación de esfuerzos por historias de usuario

La medida utilizada para la estimación del esfuerzo asociado a la implementación de cada una de las historias de usuario es el punto (máximo esfuerzo), la cual queda establecida por el equipo de desarrollo. Esta medida habitualmente toma valores de 1 a 3 en dependencia de la complejidad de la HU, partiendo de los factores que el equipo decida para clasificarlas. Estos factores por lo general son: la complejidad, el esfuerzo y el riesgo que demande cada una de ellas. Cuando una HU tiene tiempo de duración inferior a una semana significa que está muy bien descrita y que puede ser combinada con otra, y cuando tiene más de 3 indica que debemos dividirla en partes (64). 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 3.

Tabla 3: Estimación del esfuerzo por historia de usuario.

Número	Historia de usuario	Puntos de Estimación (Semanas)
1	Obtener conjunto de características	1
2	Seleccionar características relevantes	2
3	Visualizar características relevantes	1

2.5.3. Plan de iteraciones

Luego de ser identificada, descrita y estimada cada una de las HU se procede a planificar la fase de implementación del proyecto, la cual se realizará en 1 iteración. Al finalizar cada una de las funcionalidades, estas serán mostradas al cliente con el objetivo de detectar cambios necesarios o errores existentes. En la tabla 4 se muestra el plan de iteraciones para la solución propuesta.

Tabla 4: Plan de duración de iteraciones.

Iteración	Orden de la historia de usuario a implementar	Duración de la iteración(Semanas)
1	Obtener conjunto de características.	1
	Seleccionar características relevantes.	2
	Visualizar características relevantes.	1

2.5.4. Plan de entrega.

En este plan se detalla la fecha fin de cada iteración, entregando una versión desarrollada del producto hasta ese momento al cliente, en aras de lograr entregar el producto final en la fecha establecida. A continuación, se muestra el plan de entrega para cada iteración comenzando la implementación el día 6 de abril del 2017:

Tabla 5: Plan de entregas.

Iteración	Fecha de entrega
1	4/5/2017

2.6. 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 CRC.

2.6.1. Tarjetas Clase-Responsabilidad-Colaboración.

Las tarjetas CRC son utilizadas para representar las responsabilidades de las clases y sus interacciones. 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, S. & Reinaga, H. Identificación y modelado de aspectos tempranos dirigido por tarjetas de responsabilidades y colaboraciones. In XIV Congreso Argentino de Ciencias de la Computación.].

En las tablas 6 y 7 se muestran las tarjetas CRC correspondientes a las clases HormigaArtificial y Grafo.

Tabla 6: Tarjeta CRC para la clase HormigaArtificial.

Clase: HormigaArtificial	
Responsabilidad	Colaboración

<ul style="list-style-type: none"> Realizar el movimiento de las hormigas por el grafo de acuerdo a la regla de transición 	<ul style="list-style-type: none"> Hormiga Grafo
---	--

Tabla 7: Tarjeta CRC para la clase Grafo.

Clase: Grafo	
Responsabilidad	Colaboración
<ul style="list-style-type: none"> Crear y distribuir las hormigas en el grafo Inicializar y actualizar las feromonas en el grafo 	<ul style="list-style-type: none"> Rasgo Hormiga HormigaArtificial

2.6.2. 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 del producto, el mantenimiento, rendimiento y éxito global (65).

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.

2.6.3. 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 (65).

En el presente trabajo se hace uso del estilo arquitectónico modelo-vista-controlador, logrando que el sistema quede organizado y tener un orden lógico en la programación del mismo.

Modelo Vista Controlador

MVC¹⁶ es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web, donde la vista es la interfaz de usuario y el código es el que provee de datos dinámicos a la página; el modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio; y el controlador es el responsable de recibir los eventos de entrada desde la vista.

En esta investigación se presenta el sistema (Sistema para el análisis espacial en salud XANGEO) que está compuesto por los módulos:

- Común provee útiles para facilita la conexión con el SIG.
- Los módulos X_GATMa, Servicios, GeoSIMAC, y en específico Estratificador que implementa el patrón arquitectónico MVC como se muestra en la figura 2.

¹⁶ Significado para el término: Modelo Vista Controlador.

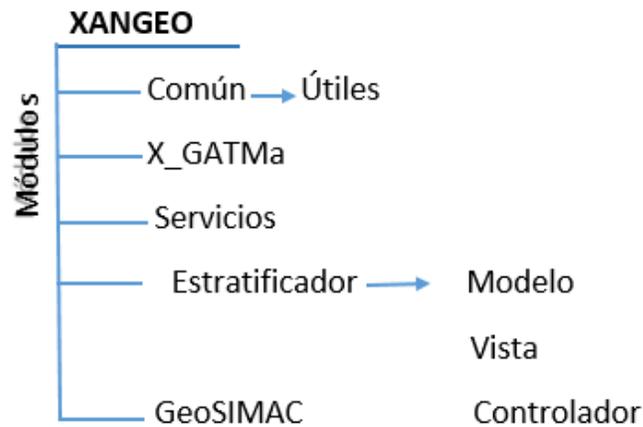


Figura 2: Evidencia de la arquitectura del sistema.

Modelo: Esta es la representación específica de la información con la cual el sistema opera. Maneja los datos y controla sus transformaciones. Este no tiene conocimiento específico de los controladores y las vistas, ni siquiera contiene referencias a ellos. Es el propio sistema el que tiene encomendada la responsabilidad de mantener enlaces entre el modelo y sus vistas y notificar a las vistas cuando cambia el modelo.

Vista: Maneja la presentación visual de los datos representados por el modelo, el cual es presentado en un formato adecuado para interactuar, usualmente la interfaz de usuario.

Controlador: Responde a eventos, usualmente acciones del usuario, e invoca cambios en el modelo y probablemente en la vista.

2.7. Patrones de diseño

Los patrones de diseño establecen la base para la búsqueda de soluciones a problemas usuales en el desarrollo de software y otros ámbitos concernientes al diseño de interacción o interfaces. Un patrón de diseño resulta ser una solución a un problema de diseño. Para que una solución sea estimada un patrón debe tener ciertas características: haber comprobado su efectividad resolviendo problemas similares en ocasiones anteriores, ser reutilizable, lo que significa que es aplicable a diferentes problemas de diseño en distintas circunstancias.

2.7.1. Patrones de diseño GRASP

Los patrones GRASP¹⁷ son utilizados para describir los principios fundamentales del diseño y la asignación de responsabilidades (66) Entre los que se utilizaron en la solución figuran los siguientes: Experto, Creador, Controlador, Bajo acoplamiento y Alta cohesión.

Experto: el patrón Experto define cómo 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 HormigaArtificial, como esta posee toda la información necesaria para analizar el movimiento de las hormigas se le atribuye dicha responsabilidad.



Figura 3: Evidencia del patrón experto.

Creador: este patrón guía la asignación de responsabilidades relacionadas con la creación de objetos. Su intención 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 Grafo, como esta crea objetos de tipo rasgo se le atribuye dicha responsabilidad.

¹⁷Siglas en inglés para el término: Patrones Generales de Software para la Asignación de Responsabilidades.

Grafo
-rasgos : Lista<Rasgo> -hormigas : Lista<Hormiga> -feromonas : Lista<entero> -evap : real -visitas : Lista<entero>
+crearHormigas() +distribuirHormigas() +actualizarFeromonas() +cantidadDeHormigas() +partition()

Figura 4: Evidencia del patrón creador.

Controlador: permite asignar la responsabilidad de controlar el flujo de eventos, 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 XangeoFSControlador.

XangeoFSControlador
-rasgos -grafo -datos -menu
+XangeoFSControlador() +extraerRasgosDeExcel() +construirGrafo() +crearHormiguero() +selection() +iteracion()

Figura 5: Evidencia del patrón controlador.

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 sean muy pocas, 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.

2.7.2. Patrones de diseño GoF

Los patrones GoF¹⁸ resuelven problemas específico de diseño de software (67) .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 (67).

Creacionales: Permiten abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados (67).

Estructurales: Describen cómo las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades (67).

Método plantilla: es un patrón de comportamiento que define en una operación el esqueleto de un algoritmo, delegando en las subclasses algunos de sus pasos, esto permite que las subclasses redefinan ciertos pasos de un algoritmo sin cambiar estructura. Este patrón se evidencia en la clase Hormiga.

¹⁸Siglas en inglés para el término: Patrones del Grupo de Cuatro.

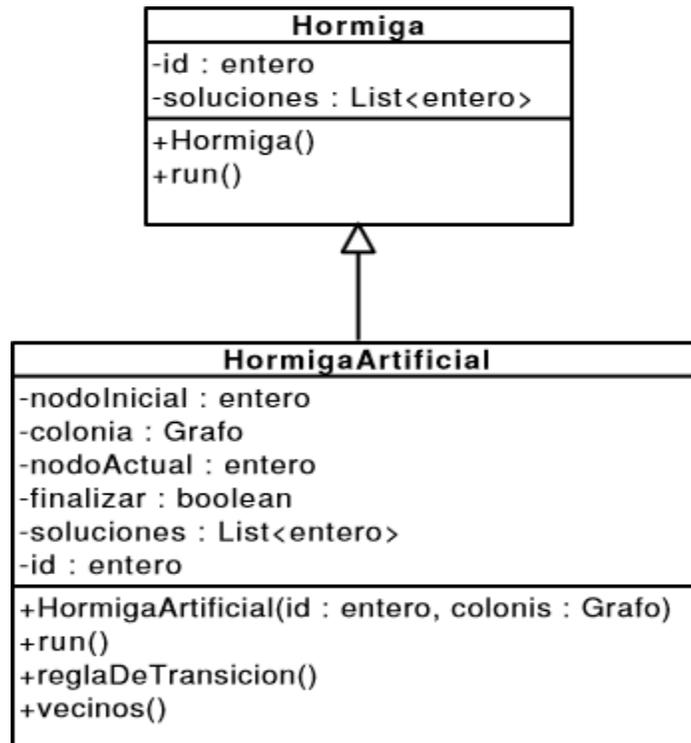


Figura 6: Evidencia del patrón plantilla.

2.8. Conclusiones parciales

En el presente capítulo se ha descrito un modelo para la selección de atributos relevantes sobre el sistema de estratificación de territorios. Como parte del mismo se ha realizado el diseño de varios algoritmos. Además se especifican los requisitos de software y los artefactos correspondientes a las fases de planificación y diseño de la metodología seleccionada. Así como la arquitectura y los principales patrones de diseño utilizados en el desarrollo de la solución. Finalmente se concluye lo siguiente:

- La propuesta de solución definida facilitó la realización de la selección de características relevantes 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 en cada una de ellas, se logró una mejor organización del trabajo y el establecimiento de fechas para la culminación por cada una de la iteración.

- El estilo arquitectónico MVC y los patrones de diseño GRASP y GoF permitió una mejor estructuración de la aplicación.
- En la implementación se detallaron las tareas de ingeniería correspondiente a cada HU, lo que permitió la organización del trabajo en una secuencia lógica de pasos.

CAPÍTULO 3. VERIFICACIÓN DE LA SOLUCIÓN PROPUESTA

En el siguiente capítulo se puntualizan las tareas de ingenierías que conforman cada HU definida en la fase de planificación. Se especifica el estándar de codificación que se estará manejando en el desarrollo de la solución. Además, se efectúan las pruebas definidas por la metodología escogida, así como las pruebas de caja blanca 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 confirmar 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 implementación o codificación 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. Tarea de ingeniería

Cada HU está compuesta por una o varias tareas de ingeniería, éstas se realizan para especificar las acciones llevadas a cabo por el programador responsable de ella o ellas. Luego de obtener las principales funcionalidades del sistema, se identificaron 3(tres) tareas de ingeniería; una por cada HU. En la tabla 8 se muestra una breve descripción de 1(una) de ellas.

Tabla 8: Tarea de ingeniería # 1.

Tarea de ingeniería			
Número Tarea:	1	Número de historia de usuario:	2
Historia de usuario:	Seleccionar características relevantes		
Tipo de Tarea:	Desarrollo	Puntos Estimados:	2

Fecha Inicio:	4/4/2017	Fecha fin:	12/6/2017
Programador Responsable:	Lisette Castro Palacio y Rosmery Hernández Peña		
Descripción:	Debe ser capaz de seleccionar las características relevantes		

3.1.2. Estándares de codificación

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

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 cada método 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 plasmar una gran cuantía de pruebas para verificar que el código esté correcto. Estas pruebas habitualmente son ejecutadas en varias ocasiones y se ven afectadas por los cambios que se introducen acorde se va construyendo la solución. XP fragmenta las pruebas en dos grupos: pruebas de aceptación, o pruebas funcionales diseñadas por el cliente final, destinadas a valorar si al final de una iteración se obtuvo la funcionalidad requerida y pruebas unitarias, encargadas de verificar el código, diseñadas por los programadores.

3.2.1. Pruebas de caja blanca

Las pruebas de caja blanca (también conocidas como pruebas de caja de cristal o pruebas estructurales) 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 garantizando que se devuelvan los valores de salida adecuados (65).

Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que:

- ✚ Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
- ✚ Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
- ✚ Ejecuten todos los bucles en sus límites operacionales.
- ✚ Ejerciten las estructuras internas de datos para asegurar su validez.

Es por ello que se considera a la prueba de caja blanca como uno de los tipos de pruebas más importantes que se le aplican a los software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad (65).

La técnica utilizada dentro de las pruebas de caja blanca fue camino básico, permite obtener una medida de la complejidad lógica de un diseño y usar esta medida como guía para la definición de un conjunto básico.

En la figura 7 se enumera la sentencia de código del método cantidadDeHormigas ().

```
def cantidadDeHormigas(self):# reglas basadas en (Kudo & Sklansky, 2000)
    if len(self.rasgos)<20:
        return len(self.rasgos)
    elif len(self.rasgos)>=20 and len(self.rasgos)<50:
        if 0.666*len(self.rasgos)<=24:
            return 24
        else:
            return 0.666*len(self.rasgos)
    else:
        if 0.5*len(self.rasgos)<=33:
            return 33
        else:
            return math.ceil(0.5*len(self.rasgos))
```

Figura 7: Código del método cantidadDeHormigas ().

A partir del grafo de flujo del método cantidadDeHormigas () que se presenta en la figura 8, la complejidad ciclomática sería:

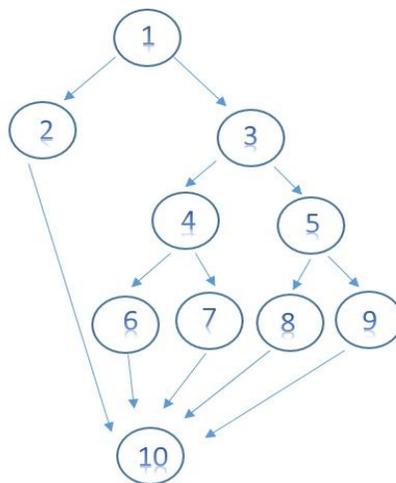


Figura 8: Grafo de flujo del método cantidadDeHormigas ().

- Como el grafo tiene 5 regiones, $V(G) = 5$
- Como el grafo tiene 3 aristas y 3 nodos, $V(G) = 13 - 10 + 2 = 5$

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

A fin de que el cálculo de las tres fórmulas anteriores se aprecia que el resultado es equivalente, representa que existen 5 posibles caminos por donde el flujo puede circular e incorpora al mismo tiempo el número mínimo de casos de pruebas para el procedimiento tratado.

Caminos básicos encontrados:

- ✚ **Camino 1:** 1-2-10
- ✚ **Camino 2:** 1-3-4-6-10
- ✚ **Camino 3:** 1-3-4-7-10
- ✚ **Camino 4:** 1-3-5-8-10
- ✚ **Camino 5:** 1-3-5-9-10

Para cada camino básico concluyente se efectúa un diseño de caso de prueba.

Tabla 9: Caso de prueba para el camino básico # 1.

Caso de prueba para el camino básico #1 (1-2-10)	
Descripción	Prueba para comprobar los resultados de la función cantidadDeHormigas () en caso que la cantidad de rasgos sean menor que 20.
Condición de ejecución	if len(self.rasgos)<20
Entrada	len(self.rasgos)
Resultado	len(self.rasgos)
Resultado de la prueba	Prueba Satisfactoria

Tabla 10: Caso de prueba para el camino básico # 2.

Caso de prueba para el camino básico #2 (1-3-4-6-10)

Descripción	Prueba para comprobar los resultados de la función cantidadDeHormigas() en caso que la cantidad de rasgos este en el rango de 20 y 50 y es menor o igual que 24
Condición de ejecución	len (self. rasgos)>=20 and len (self. rasgos) <50: if 0.666*len(self.rasgos)<=24:
Entrada	len(self.rasgos)
Resultado	24
Resultado de la prueba	Prueba Satisfactoria

Tabla 11: Caso de prueba para el camino básico # 3.

Caso de prueba para el camino básico #3 (1-3-4-7-10)	
Descripción	Prueba para comprobar los resultados de la función cantidadDeHormigas() en caso que la cantidad de rasgos este en el rango de 20 y 50 y es mayor que 24
Condición de ejecución	len (self. rasgos)>=20 and len (self. rasgos) <50: if 0.666*len(self.rasgos)>24:
Entrada	len(self.rasgos)
Resultado	return 0.666*len(self.rasgos)
Resultado de la prueba	Prueba Satisfactoria

Tabla 12: Caso de prueba para el camino básico #4.

Caso de prueba para el camino básico #4 (1-3-5-8-10)	
Descripción	Prueba para comprobar los resultados de la función cantidadDeHormigas() en caso que la cantidad de rasgos este en el rango de 20 y 50 y es menor o igual que 33
Condición de ejecución	len (self. rasgos)>=20 and len (self. rasgos) <50: if 0.5*len(self.rasgos)<=33:
Entrada	len(self.rasgos)
Resultado	33
Resultado de la prueba	Prueba Satisfactoria

Tabla 13: Caso de prueba para el camino básico #5.

Caso de prueba para el camino básico #5 (1-3-5-9-10)	
Descripción	Prueba para comprobar los resultados de la función cantidadDeHormigas ().Si $20 < \text{cantidad de rasgos} > 50$ y es mayor que 33.
Condición de ejecución	len (self. rasgos)>=20 and len (self. rasgos) <50: if 0.5*len(self.rasgos)<=33:
Entrada	len(self.rasgos)
Resultado	return math.ceil(0.5*len(self.rasgos))

Resultado de la prueba	Prueba Satisfactoria
-------------------------------	-----------------------------

Análisis de los resultados

Para la validación del código generado en el desarrollo de la herramienta se escogieron los métodos más relevantes, a los que se le efectuaron las pruebas para poder evaluar si el funcionamiento de cada uno de estos métodos se comportó de la manera esperada. Se realizaron un total 11 pruebas a las 4 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, evidenciándose la estabilidad de la lógica aplicada en el código generado en el desarrollo de la herramienta informática.

3.2.2. Pruebas de aceptación

El objetivo de estas pruebas es verificar los requisitos, por este motivo, los propios requisitos del sistema son la principal fuente de información a la hora de construir las pruebas de aceptación, las cuales son creadas a partir de las historias de usuario, durante una iteración la historia de usuario seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación.

El cliente o usuario especifica los aspectos a testear cuando una historia de usuario ha sido correctamente implementada. Una historia de usuario puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento y no se considera completa hasta que no supera sus pruebas de aceptación. Esto significa que debe desarrollarse un nuevo test de aceptación para cada iteración o se considerará que el equipo de desarrollo no realiza ningún progreso (68).

Una prueba de aceptación es como una caja negra, cada una de ellas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección de las pruebas de aceptación y tomar decisiones acerca de las mismas. La garantía de calidad es una parte esencial en el proceso de XP. La realización de este tipo de pruebas y la publicación de los resultados debe ser los más rápido posibles, para que los desarrolladores puedan realizar con la mayor rapidez los cambios que sean necesarios.

Tabla 14: Caso de prueba de aceptación seleccionar características relevantes.

Caso de prueba de aceptación	
Código: HU2	Historia de Usuario: 2
Nombre: Seleccionar características relevantes.	
Descripción: Prueba para validar la funcionalidad seleccionar características relevantes.	
<p>Condiciones de ejecución:</p> <ul style="list-style-type: none"> • El sistema tiene la capa de territorios a evaluar, seleccionada por el usuario. • El sistema cuenta con los datos de los indicadores estadísticos y/o los datos de los indicadores cartográficos. • El usuario debe seleccionar la opción Estratificar. • El usuario debe seleccionar un algoritmo de agrupamiento. • El usuario debe seleccionar al menos tres territorios a evaluar. • El usuario debe seleccionar el nivel geográfico de los territorios a evaluar. • 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 Opciones Avanzadas). • El usuario debe seleccionar la opción de selección automática. • El usuario debe seleccionar un criterio de riesgo de salud para cada indicador elegido. • El usuario debe seleccionar la opción Aceptar. 	
Resultados esperados: 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.	
Evaluación de la prueba: Prueba satisfactoria.	

3.2.3. Caso de estudio

Con el fin de estimar los efectos de la solución propuesta se decide aplicar a un caso de estudio, en correspondencia con el trabajo realizado por (10) 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 (69) 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 al caso de estudio

Para realizar la estratificación de las provincias de Cuba, se utilizó la herramienta existente aplicando la selección de características relevantes y se obtuvo el resultado que se muestra en la figura 9.

The screenshot shows the 'Estratificar' application window. It is divided into several sections:

- ALGORITMOS DE AGRUPAMIENTO:** Radio buttons for 'K-MEDIAS' (selected) and 'MATRIZ DE SIMILITUD'. A 'Datos' dropdown menu is below.
- TERRITORIOS:** A list of territories to evaluate: Matanzas, Villa Clara, Cienfuegos, Pinar del Ro, La Habana, and Ciudad de la Hbana. All are checked. A 'Seleccionar Todos' button is below. Below this, a radio button for 'PROVINCIA' is selected over 'MUNICIPIO'.
- INDICADORES A EVALUAR:** A sub-section for 'INDICADORES ESTADÍSTICOS' with 'Selección automática' checked. A list includes 'densidad poblacional' (checked), 'natalidad', 'población menor de 15 años' (checked), 'población mayor de 60 años' (checked), and 'población menor de 1 año'.
- INDICADORES CARTOGRÁFICOS:** A sub-section with 'Seleccionar Todos' unchecked.
- CRITERIOS PARA EVALUAR EL RIESGO DE LOS INDICADORES SELECCIONADOS:** A table with columns 'Indicadores', 'A Mayor Valor Mayor el Riesgo', and 'A Mayor Valor Menor el Riesgo'.

	Indicadores	A Mayor Valor Mayor el Riesgo	A Mayor Valor Menor el Riesgo
1	densidad poblacional	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	población menor de 15 años	<input type="checkbox"/>	<input checked="" type="checkbox"/>
3	población mayor de 60 años	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	mortalidad infantil	<input checked="" type="checkbox"/>	<input type="checkbox"/>
5	mortalidad materna	<input checked="" type="checkbox"/>	<input type="checkbox"/>
6	incidencia de henatitis	<input checked="" type="checkbox"/>	<input type="checkbox"/>
- INDICADORES CARTOGRÁFICOS:** A table with columns 'Indicadores', 'A Mayor Valor Mayor el Riesgo', and 'A Mayor Valor Menor el Riesgo'. It is currently empty.

Buttons for 'Opciones Avanzadas', 'Aceptar', and 'Cancelar' are visible.

Figura 9: Interfaz de usuario VistaEstratificador.

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 10 se presenta en forma de mapa y en la tabla 15 de manera más detallada.

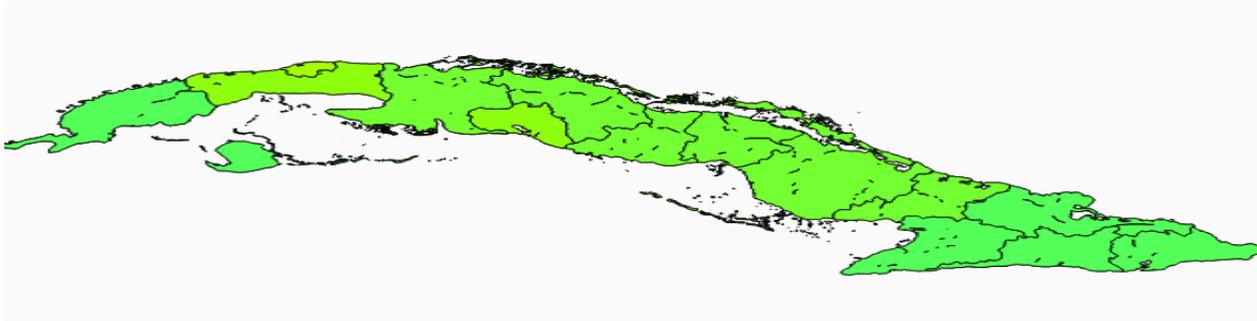


Figura 10: Mapa temático de la estratificación realizada con la selección de características relevantes.

Tabla 15: Resultados de la estratificación realizada con la selección de características relevantes.

Nombre del estrato	Provincias	Riesgos de salud
Estrato 1	Ciudad de la Habana	0.51
Estrato 2	Villa Clara, Las Tunas, Camagüey ,Sancti Espíritus, Ciego de Ávila	0.43
Estrato 3	Cienfuegos, Matanzas, La Habana	0.47
Estrato 4	Pinar del Río, Granma, Holguín , Santiago de Cuba, Guantánamo	0.38

Comparación de resultados

Para determinar la similitud existente entre los resultados obtenidos aplicando la selección de atributos relevantes y la forma de efectuar el proceso realizado en (70) (10) 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 3 (71).

$$r = \frac{\sum_{i=1}^k a_i}{n} \quad (3)$$

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, aquella que tenga el número máximo de objetos bien clasificados. 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 4:

$$e = 1 - r \quad (4)$$

Una medida de error en un rango ≤ 0.3 , representa un valor aceptable de error (71).

A continuación se presenta una tabla comparativa entre los resultados del proceso de estratificación realizado utilizando la selección de atributos relevantes y como se realizaba el proceso anteriormente en (70).

Tabla 16: Tabla comparativa de los resultados de los procesos de estratificación realizados (por estratos).

Número de estratos (i)	Resultados obtenidos por Bombino (10) 1	Resultados obtenidos por Rolando y Yanisley (70) 2	Resultados con los atributos relevantes 3	Territorios clasificados correctamente con respecto entre :	
				1(10) y 3	2(68) y 3(10)
1	Ciudad de la Habana	Ciudad de la Habana	Ciudad de la Habana	1	1
2	La Habana, Matanzas, Camagüey Sancti Spíritus, Pinar del Río, Guantánamo,	La Habana, Matanzas, Cienfuegos, Camagüey, Guantánamo.	Cienfuegos, Matanzas, Camagüey, La Habana.	3	4
3	Holguín, Las Tunas,	Pinar del Río,	Pinar del Río,	2	4

	Villa Clara, Granma	Santiago de Cuba, Ciego de Ávila, Granma	Granma, Santiago de Cuba, Guantánamo, Ciego de Ávila.		
4	Cienfuegos, Santiago de Cuba, Ciego de Ávila	Holguín, Las Tunas, Villa Clara, Sancti Spíritus	Villa Clara, Las Tunas, Sancti Espíritus, Granma, Holguín.	4	4
$\sum_{i=1}^k a_i$				10	13

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, entre **1** y **3**; mientras se obtuvo un error de 0,07142 entre **2** y **3**. 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 realizar la selección de características relevantes a partir del caso de estudio propuesto.

Los resultados expuestos demuestran la utilidad y validez del empleo de la selección de atributos relevantes en el proceso de estratificación territorial, permitiendo obtener información precisa en estudios salubristas, eliminando la gran cantidad de información tanto irrelevante como redundante que dificulta la posibilidad de brindar información precisa.

3.3. Conclusiones parciales

Luego del trabajo realizado se arribó a las siguientes conclusiones:

- Las tareas de ingeniería correspondiente a cada HU, permitió la organización del trabajo en una secuencia lógica de pasos.

- El estándar de codificación utilizado facilitó el entendimiento del código y una mejor organización del mismo. Las pruebas de aceptación y de caja blanca efectuadas proporcionaron 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 permitió 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 con mayor grado de información precisa, eliminando la gran cantidad de información redundante e irrelevante.
- 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

Como resultados de la presente investigación se obtuvo un componente para la selección de características relevantes en el proceso de estratificación de territorios, que contribuye a la obtención de información precisa en estudios salubristas. En función de los resultados obtenidos se arribó a las siguientes conclusiones:

- La definición del marco teórico referencial de la investigación relacionada con el proceso de estratificación de territorios, fundamentó la necesidad de desarrollar un componente que sea capaz de seleccionar características relevantes, se adapte a los objetivos expuestos y satisfaga las necesidades del país.
- La utilización de la metaheurísticas ACO en el diseño propuesto, aunque no garantiza una solución óptima, si obtiene una solución de calidad, con un costo computacional aceptable a pesar de estar referenciado el problema de la selección de características relevantes como NP.
- La integración de la propuesta al SIG Qgis facilitó la realización del proceso de estratificación de territorios con un costo computacional adecuado y manteniendo una calidad aceptable en los estratos en correspondencia a otras investigaciones realizadas.
- Las pruebas aplicadas para la verificación de la solución informática y la valoración de los resultados a través de un caso de estudio evidenció la efectividad del modelo propuesto, avalando su correcto funcionamiento.

RECOMENDACIONES

Se recomienda para posteriores versiones que se incluyan elementos como:

- Diseñar otras estrategias de selección del subconjunto de características relevantes a partir de la salida del algoritmo propuesto y comparar los resultados utilizando medidas de calidad.
- Incorporar medidas para evaluar la calidad del subconjunto relevante seleccionado a partir de los referentes que provee la teoría de conjuntos aproximados.
- Proponer otros algoritmos de selección de características relevantes basadas en otros metaheurísticas y evaluar su comportamiento a partir de su costo computacional y la calidad de la selección.

REFERENCIAS BIBLIOGRÁFICAS

1. **al., Suprem et, y otros.** A review on application of technology systems, standards and interfaces for agriculture and food sector. *Computer Standards & Interfaces*. 2013.
2. **Wang, Y.** Meteoinfo: Gis software for meteorological data visualization and analysis. *Meteorological*. 2014.
3. **Olaya, VÍCTOR.** Sistemas de Información Geográfica. 2011, Vol. 1.0.
4. **Tanser, F C y Le Sueur, D.** The application of geographical information systems to important public health problems in Africa. *International journal of health geographics*,. 2002.
5. **Shi, X y Kwan, M.** Introduction: geospatial health research and gis. *Annals of GIS*. 2015.
6. **Olaya, Victor.** *Sistemas de Información Geográfica*. 2011. Vol. 1.0.
7. **BRAVO, J Domínguez.** Breve introducción a la cartografía y a los sistemas de información geográfica (SIG). *Ciemat*. [En línea] 2000. [Citado el: 20 de mayo de 2016.] <http://hc.rediris.es/pub/bscw.cgi/d251342/itcsig.pdf>.
8. **Rojas y Iñiguez , Luisa.** Geografía y salud: temas y perspectivas en América Latina Geography and health: themes and perspectives in Latin America. 1998. Vol. 14, 4.
9. **Batista Moliner, Dr. Ricardo, y otros.** *Guía para la realización del proceso de estratificación epidemiológica*. La Habana: MINSAP. : s.n., 2001.
10. **Companioni, Yenisei Bombino.** Tipología Estructural Articulada. *metodología de análisis para la estratificación según indicadores de salud*. [En línea] 2005. [Citado el: 21 de marzo de 2016.] http://bvs.sld.cu/uats/rtv_files/2005/bombino.htm.
11. **H, Lui y Motoda.** Feature Selection: An Ever Evolving Frontier in Data Mining. [En línea] 2007.
12. **Fotheringham y Rogerson.** Spatial analysis and GIS. *CRC Press*. 2013.
13. **Sarría, Francisco Alonso.** *Sistemas de Información Geográfica*. 2014.
14. **Martínez, Lorenzo.** Cartografía, urbanismo y desarrollo inmobiliario. s.l. : Dossat, 2000.
15. *Servicio de mapas temáticos. Mapping.* **Torres Rodríguez, Alexander y Puente Rodriguez, Rafael.** 139, 2010.
16. **TEAM, GRASS DEVELOPMENT.** The world's leading Free GIS software. [En línea] 2015. [Citado el: 20 de mayo de 2016.] <http://grass.osgeo.org/>.
17. **Böhner, J, y otros.** System for automated geoscientific analyses. *System for Automated Geoscientific Analyses*. [En línea] 2015. [Citado el: 3 de enero de 2016.] <http://www.saga-gis.org/en/index.html> .
18. **TEAM, QGIS DEVELOPMENT.** QGIS A Free and Open Source Geographic Information System. [En línea] 2012. [Citado el: 23 de marzo de 2016.] <http://www.qgis.org/en/site/>.
19. **Dudani, S.** The Distance-Weighted k-Nearest-Neighbor Rule. *Man and Cybernetics*. 1975.
20. *Useful Feature Subsets and Rough Set Reducts. Paper Presented at the 3rd Int. Workshop on Rough Set and Soft Computing.* . **KOHAVI, R., & FRASCA, B.** 1994.
21. **YANG, J y HONAVAR, V.** Feature Extraction. *Construction and Selection*. 1998. págs. 117-136.
22. **BREIMAN, L., FRIEDMAN, J., OLSHEN, R., & STONE, C.** Classification and Regression Trees. *Paper Presented at the Wadsworth Int. Group*. 1984.
23. **LIU, H y YU, L.** Feature Selection for data Mining. . Arizona : s.n., 2002.
24. **DAVIES, S., & RUSSELL, S.** Completeness of Searches for Smallest Possible Feature Sets. *Paper Presented at the AAAI Fall Symposium on Relevance.* . 1994.

25. **SIEDLECKI , W y SKLANSKY, J.** On Automatic Feature Selection. *Int. Journal of Pattern*. 1988. págs. 197-220.
26. **LIU, H y YU, L.** Toward Integrating Feature Selection Algorithms for Classification and Clustering . *IEEE Trans. On Knowledge and data Enginnering*, 17, p. 1-12. 2005. 2005. 17.
27. **RUIZ, D. R.** Selección de Atributos mediante proyecciones. *Unpublished Tesis Doctoral*. Universidad de Sevilla, Sevilla : s.n., 2005.
28. **BELL, D y WANG, H.** Formalism for Relevance and its Application in Feature Subset Selection. *Machine Learning*. 2000.
29. **BLUM, A y LANGLEY, P.** Selection of Relevant Features and Examples in Machine Learning. *Artificial Intelligence*. 1997.
30. *Selección de atributos relevantes aplicando algoritmos que combinan conjuntos aproximados y optimización en colonias de hormigas.* **Rodríguez, Yanela, y otros.** 1, La Habana : s.n., 2014, Vol. 8.
31. **Dorigo, Marco y Stutzle.** *Ant Colony Optimization*. 2004.
32. **Gómez Díaz, MSc. Yudel Roderigo, Bello Pérez, Dr. Rafael Esteban y Nowé, Dra. Ann .** Algoritmos que combinan conjuntos. *Algoritmos que combinan conjuntos*. 2010.
33. **JENSEN, R, & SHEN, Q.** Finding Rough Set Reducts with Ant Colony Optimization. *Paperpresented at the UK Workshop on Computational Intelligence*. 2003.
34. **Tema-2.pdf.** [En línea] abril de 2016. [Citado el: 5 de 15 de 2017.] [.\(https://oposinet.cvexpres.com/wp-content/uploads/2016/04/Tema-2.pdf\)](https://oposinet.cvexpres.com/wp-content/uploads/2016/04/Tema-2.pdf).
35. *Aplicación de las gramáticas de grafo en Sistemas de Información Geográfica.* **Rodríguez Puente, Rafael.** La habana : Revista Cubana de Ciencias Informáticas, 2010.
36. **Rodríguez Corvea, L.** Detención de conglomerados en la solución de problemas bioinformáticos y biomédicos. . *PhD thesis, Universidad Central "Marta Abreu" de las Villas.* . s.l. : Facultad de Matemática, Física y Computación, 2001.
37. **Duda, R O y Hart, P E.** Pattern classification and scene analysis. *Pattern classification and scene analysis*. Wiley New York : s.n., 1973, Vol. 3.
38. **Duda., R O, Hart, P E y Stork, D G.** Pattern classification. [aut. libro] John Wiley & Sons. *Pattern classification*. 2012.
39. **Dempster, A P, Laird, N M y Rubin, D B.** Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, . 1977, págs. 1-38.
40. **Fukunaga, K.** Introduction to statistical pattern recognition. . Academic press : s.n., 2013.
41. **Jain, A. K. & Flynn, P. J.** Image segmentation using clustering. NJ : IEEE Press, Piscataway, 1996.
42. **Jain, A.K. y Duin, R.P.&Mao,J.** Statistical pattern recognition:Areview. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*. 2000, págs. 4–37.
43. **Salton, G.** Automatic term class construction using relevance—a summary of work iautomaticpseudoclassification. *InformationProcessing&Management*. 1980, págs. 1–15.
44. **Zhai, C. & Lafferty, J.** A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems (TOIS)*. 2004, págs. 179–214.
45. **Cutting, D. R., Karger, D. R. y Pedersen, J. O. & Tukey, J. W.** Scatter/gather: A cluster-basedapproachtobrowsinglargedocumentcollections. *In Proceeding softhe 15th annual international ACMSIGIR conference on Research and development in information retrieval*. 1992, págs. 318–329.
46. **Dhillon, I. S. y Mallela, S. & Kumar, R.** Enhanced word clustering for hierarchical text classification. *In Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2002, págs. 191–200.

47. Xu, X., Ester, M. y Kriegel, H.-P. & Sander, J. A distribution-based clustering algorithm for mining in large spatial databases. *In Data Engineering*. 14th International Conference : s.n., 1998, págs. 324–331.
48. Ester, M., Frommelt, A. y Kriegel, H.-P. & Sander, J. Spatial data mining: database primitives, algorithms and efficient DBMS support. *Data Mining and Knowledge Discovery*. 2000, págs. 193–216.
49. *La minería de datos espaciales y su aplicación en los estudios de salud y epidemiología*. González Polanco, L. & Pérez Betancourt, G. La Habana : s.n., 2013, Revista Cubana de Información en Ciencias de la Salud, págs. 482–489.
50. Heer, J. & Chi, E. H. Identification of web user traffic composition using multi-modal clustering and information scent. *In Proc. of the Workshop on Web Mining, SIAM Conference on Data Mining*. 2001, págs. 51–58.
51. Foss, A. y Wang, W. & Zaïane, O. R. A non-parametric approach to web log analysis. *In Proc. of Work shop on Web Mining in First International SIAM Conference on Data Mining*. 2001, págs. 51–58.
52. —. A non-parametric approach to web log analysis. *Proc. of Workshop on Web Mining in First International SIAM Conference on Data Mining*. 2001, págs. 41–50.
53. Xu, Y. y Olman, V. & Xu, D. Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees .*Bioinformatics*. 2002, págs. 536–545.
54. González, D. P. Algoritmos de agrupamiento basados en densidad y variación de clusters. PhD thesis, Universitat Jaume I, Departament de Llenguatges i Sistemes Informàtics. 2010.
55. Schefer, Wenzl, Sobernig y Strembeck. [*SchefEvaluating A Uml-Based Modeling Framework For Process-Related Security Properties: A Qualitative Multi-Method Study*. .
56. Gonzalez en Prezi, Miguel Angel. HERRAMIENTAS CASE. [En línea] 2013. [Citado el: 29 de mayo de 2016.] https://prezi.com/aad8mbta_vjb/herramientas-case/ .
57. Software Design Tools for Agile Teams, with UML, BPMN and More.
58. Letelier, P. Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP). [En línea] 2006. [Citado el: 20 de mayo de 2016.] http://www.cyta.com.ar/ta0502/b_v5n2a1.htm .
59. Joskowicz , J. Reglas y prácticas en eXtreme Programming . Universidad de Vigo. Universidad de Vigo : s.n., 2008.
60. Beck, K. Extreme programming explained: embrace change. [En línea] 2000. [Citado el: 20 de mayo de 2016.] <http://books.google.es/books?hl=es&lr=&id=G8EL4H4vf7UC&oi=fnd&pg=PR13&dq=Extreme+Programing>.
61. *An unsupervised feature selection algorithm based on ant colony optimization*. Tabakhi , Sina y Moradi, Parham . Junio de 2014.
62. Kudo , y otros. Comparison of algorithms that select features for pattern classifiers. 2000.
63. Sommerville, I y Galipienso, M. Ingeniería del software. [En línea] 2005. [Citado el: 23 de marzo de 2016.] <http://books.google.es/books?hl=es&lr=&id=gQWd49zSut4C&oi=fnd&pg=PA1&dq=Ingenieria+de+Software+lan+Sommerville&ots=s623rrszwd&sig=DVAJiDo0wOzveatAqj48nPXvtsl> .
64. Ordoñez Pérez, Mayelin y Rodríguez Corbea, Maite . 2007.
65. Pressman, R. *Ingeniería del software. Un enfoque práctico*. SA de CV Mexico : I McGraw-Hill, 2005. Sexta edición.

66. Larman, C. UML y patrones. introducción al análisis y diseño orientado a objetos. [En línea] Pearson, 1999. [Citado el: 27 de mayo de 2016.] <http://gravepa.com/granaino/biblioteca/aprende/UNED%20-%20Grado%20Inform%C3%A1tica%20-%20Extras%20%26%20Ediciones%20Antiguas/Una%20introduccion%20al%20an%C3%A1lisis%20y%20dise%C3%B1o%20or.pdf>.
67. Rodríguez, Eric. Patrones gof. [En línea] 2010. [Citado el: 6 de enero de 2017.] http://www.academia.edu/5903473/Patrones_gof.
68. Rodríguez Corbea , Maité y Ordoñez Pérez, Mayelin. 2007.
69. INFOMED. Anuario Estadístico de Cuba . *Biblioteca Virtual en Salud de Cuba*. [En línea] [Citado el: 20 de marzo de 2017.] <http://bvscuba.sld.cu/anuario-estadistico-de-cuba/> .
70. Torres Vega. , Yanislay y Morales Pérez, Rolando. *Propuesta para la estratificación de territorios basada en indicadores de salud*. La Habana : s.n., 2015.
71. Hernández Valadez, EDNA. Algoritmo de clustering basado en entropía para descubrir grupos en atributos de tipo mixto. [En línea] 2006. [Citado el: 14 de febrero de 2017.] <http://webserver.cs.cinvestav.mx/TesisGraduados/2006/tesisEdnaHernandez.pdf>.