

**Universidad de las Ciencias Informáticas
Facultad 3**



***“Solución informática para la localización de servicios
utilizando diagramas de Voronoi”***

**Trabajo de Diploma para optar por el título de Ingeniero en
Ciencias informáticas**

Autor(es): Bárbaro K. Casanova Pérez
Carlos José Castillo Piñeiro

Tutor(es): Yadian Guillermo Pérez Betancourt
Liset González Polanco

Co-tutor: Jorge Jesús Hidalgo Ruíz

La Habana. Junio de 2015.
“Año 57 de la Revolución”.



(...) vale, pero millones de veces más la vida de un solo ser humano, que todas las propiedades del hombre más rico de la tierra (...)

Ernesto «Che» Guevara

Discurso en el acto de inauguración del curso de adoctrinamiento organizado por el Ministerio de Salud Pública el 20 de agosto de 1960.

Declaración de autoría

DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año 2015.

Autor
Bárbaro Kenier Casanova Pérez

Autor
Carlos José Castillo Piñeiro

Tutor
Ing. Liset González Polanco

Tutor
Ing. Yadian Guillermo Pérez Betancourt

Co-Tutor
Ing. Jorge Jesús Hidalgo Ruíz

DATOS DE CONTACTO

Autores:

- **Nombre y apellidos:** Bárbaro Kenier Casanova Pérez
- **Email:** bkcasanova@estudiantes.uci.cu
- **Institución:** Universidad de las Ciencias Informáticas (UCI)
- **Dirección:** Carretera a San Antonio de los Baños, Km 2 1/2, Reparto Torrens, Boyeros

- **Nombre y apellidos:** Carlos José Castillo Piñeiro
- **Email:** cjcastillo@estudiantes.uci.cu
- **Institución:** Universidad de las Ciencias Informáticas (UCI)
- **Dirección:** Carretera a San Antonio de los Baños, Km 2 1/2, Reparto Torrens, Boyeros

Tutores:

- **Nombre y apellidos:** Ing. Liset González Polanco
- **Email:** lgpolanco@uci.cu
- **Situación laboral:** Profesora
- **Institución:** Universidad de las Ciencias Informáticas (UCI)
- **Dirección:** Carretera a San Antonio de los Baños, Km 2 1/2, Reparto Torrens, Boyeros

- **Nombre y apellidos:** Ing. Yadian Guillermo Pérez Betancourt
- **Email:** ygbetancourt@uci.cu
- **Situación laboral:** Profesor
- **Institución:** Universidad de las Ciencias Informáticas (UCI)
- **Dirección:** Carretera a San Antonio de los Baños, Km 2 1/2, Reparto Torrens, Boyeros

Co-Tutor:

- **Nombre y apellidos:** Ing. Jorge Jesús Hidalgo Ruíz
- **Email:** jjperez@uci.cu
- **Situación laboral:** Profesor
- **Institución:** Universidad de las Ciencias Informáticas (UCI)
- **Dirección:** Carretera a San Antonio de los Baños, Km 2 1/2, Reparto Torrens, Boyeros

Dedicatoria

Dedico este trabajo a la única persona que no me ha, ni va a abandonarme nunca MI MAMÁ, a mi hermana para que le sirva como meta a superar, a Mami y a Papi que aunque no se encuentre entre nosotros fue un padre para mí y espero que este orgulloso de mí y a mis tíos Reinaldo, Reidel y el Chino.

También se la dedico a mi papá, que de estar entre nosotros, estoy seguro que estaría aquí apoyándome, a Maye quien fue otra madre para mí en preuniversitario y a ella le debo el ver llegado a esta universidad.

Bárbaro

Quiero dedicar este trabajo y esfuerzo a quienes nunca se han rendido conmigo y no me abandonan, a los que siempre me guían y me levantan:

A mi madre y a mi padre, a la memoria de mi tía abuela Alicia, a mis hermanos, a mi tía Silvia, a mis primos, a mis sobrinitos y a todos los que siempre creyeron en mí.

Carlos José

Agradecimientos

''Nunca es demasiado agradecimiento, a quien no te abandonó en tus peores momentos.''

Les agradezco en primer lugar a la persona que me trajo a este mundo MI MÁMA, a mis abuelos, a mi linda hermana Yenia, a mis tíos Reinaldo, El Chino y Reidel, y a toda mi familia (Si algún día leen esto, entiendan que son muchos y no los podía mencionar a todos) por la educación y enseñanzas que me dieron.

De forma general agradecerles a todas las personas que creyeron en mí y a las que dudaron de mi capacidad para enfrentar este reto, porque sin querer me dieron más fuerza para seguir adelante.

Específicamente le agradezco y lo hago de forma cronológica a Maritza y Maida mis maestras de la primaria, que todavía lo siguen siendo, a Yudit quien me soportó durante 3 años en la secundaria con mucha paciencia y dedicación.

Un agradecimiento muy especial a Maye que es la persona que me ayudó e inspiró a venir a esta Universidad. A mis amigos de la carrera: Alejandro, Vladimir, Aroldo, Yilian, Leydis, Leanet, Yoelvis, Maria, Ledian, El tabo, Victor, El ruso Chubuskova, Yaimaris, Renedis, Juan A, Débora, Arianna, Sosa, Luis Angel, Roja, Luis Javier, Roberto, Abel Alexis, Yusleidy, Elizabeth que me sirvió de testigo en todas las comisiones, a Nieve, Cueva, Marisleydis, Fredi, Elaidis, Yordan, Manuelita, Boris, Mireya, Yesenia, Los Chuchumecos, a los integrantes de la Limonada, a los compañeros(as) del Taekwondo y el fútbol Rugby, al equipo de sonido de la UCI en especial a Maikel, a los ingenieros y profes Dariela, Rosalina, Nela, Yorlen, Widen, El tiempo, Adrian, El leo, Maylevis, Yarenis, Aloy, Leandro, a las tías del comedor por darme un poquito más del pan de cada día, a los compañeros(as) de servicio del docente, al piquete de la Kizomba, al equipo de Balonmano.

También quiero agradecerle a mis amistades del pueblo donde nací y me crie por el apoyo que siempre me dieron, a mi tía Nidia, a mis Hermanos Dayron y Popo, a Bololo, Diuber, Chaleco, a Mamita y Pipo.

Un agradecimiento muy especial al Lea y a Javier y a todo el piquete de su proyecto, que fueron mis pilares en ese primer año que tanto trabajo me costó. A mis tutores que creyeron en mí y me ayudaron a conseguir este título que tanto desee.

Y para el final y uno de los más importante le agradezco a mi compañero de tesis que sin él no hubiese llegado hasta aquí y gracias a él puedo estar leyendo esto hoy. Gracias mi hermano de todo corazón.

''Hay momentos en la vida que son especiales por si solos. Compartirlo con las personas que quieres, los convierte en mágicos e inolvidables. Gracias por estar aquí.''

Bárbaro

Agradecimientos

A mis padres por ser la fuente de apoyo constante e incondicional en toda mi vida.

A mis hermanos que han sido un ejemplo a seguir

A mi tía Silvia que ha sido en estos cinco años madre, padre y amiga

A mis primos Rigo y Maiquel que siempre han estado ahí cuando los he necesitado.

A toda mi familia que de una forma u otra me han apoyado.

A todos mis amigos que siempre han creído en mí.

Especialmente agradecer a mis tutores Yadian y Liset por su esfuerzo y dedicación.

Al Ing. Luis Eduardo Benítez Oliva por ayudarme en mi tesis incondicionalmente.

A mi compañero de tesis, que ha enfrentado conmigo este gran desafío.

A mis compañeros y amigos de la UCI: Luis Angel que ha sido un hermano para mí, a Roberto por su apoyo incondicional, a Yunier mi eterno rival del Mario Kart, a Luisky mi padrino, a Yunierki, Leonardo, Yasiel, Rolando, los Abel, Dito, Luis Javier, Dayana, Rojas, Danny, Choy, Rayco, Dachel, las Elizabeth, Manuela, Gabriel, Marisleydi, Chandý, Samper, Adrian, Sheila, Yuliet, Dayanis, Rosalía, Asprón, Carlos, Yuleidis, Alfredo, Anelo, Jorge, Delisle y Alianna. A todos los que durante los 5 años de mi carrera han compartido buenos y malos momentos conmigo, sobrepasando innumerables obstáculos.

A todos los maestros y profesores que me han guiado durante toda mi vida estudiantil.

Carlos José

RESUMEN

La evolución de las tecnologías de la información y las comunicaciones, ha tenido un gran impacto en la economía mundial, particularmente en el sector terciario de la economía o sector de servicio. En este trabajo se analiza el problema de la localización de servicios, haciendo énfasis en los servicios de salud en Cuba. Se caracterizan los Sistemas de Información Geográfica (SIG), teniendo en cuenta las facilidades que estos brindan a la hora de realizar la localización y selección en el espacio. Además se identificaron las limitaciones que presentan los SIG para integrar las variables que inciden sobre los servicios. Se presenta una solución algorítmica utilizando diagramas de Voronoi y variables socioeconómicas y demográficas para la solución al problema planteado en el sector de la salud.

Palabras claves: Diagramas de Voronoi, Localización de servicio, Sistemas de Información Geográfica.

COMPUTING SOLUTION FOR THE LOCATION OF SERVICES USING VORONOI DIAGRAMS.

ABSTRACT

The evolution of the Information and Communication Technologies has had a great impact on world economy, particularly in the tertiary sector of economy or services sector. In this work, an analysis of the localization of the services, with emphasis on health services in Cuba, is done. The Geographic Information Systems (SIG), due to the possibilities they offer at the time to carry on the localization and selection of the space, are characterized. Furthermore, the limitations that SIG present for the integration of variables that influence over the services are identified. An algorithmic solution that uses Voronoi diagrams and socio-economic and demographic variables for the solution of the stated problems in the health sector is also presented.

Keywords: Geographic Information Systems, Location Service, Voronoi diagrams.

Índice de contenido

ÍNDICE

INTRODUCCIÓN	1
CAPÍTULO 1. EL PROBLEMA DE LA LOCALIZACIÓN Y REGIONALIZACIÓN DE SERVICIOS.....	5
1.1 Conceptos asociados al dominio del problema.....	5
1.2 Caracterización del análisis espacial basado en métodos de interpolación.....	7
1.3 Caracterización de los métodos de interpolación	8
1.4 Análisis de las soluciones existentes	12
1.5 Metodología de desarrollo de software.....	13
1.6 Herramientas y lenguajes	15
1.6.1 Lenguaje Unificado de Modelado (UML) 2.0	15
1.6.2 Herramientas CASE.....	16
1.6.3 Visual Paradigm for UML Enterprise Edition 8.0	16
1.6.4 Lenguajes de programación.....	17
1.6.5 Entorno de Desarrollo Integrado (IDE).....	18
1.6.6 Sistema de Información Geográfica Quantum GIS (QGIS) 2.6	18
Conclusiones del capítulo.....	19
CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.....	20
2.1 Propuesta de solución.....	20
2.2 Complejidad del algoritmo.....	23
2.3 Modelo de dominio	24
2.3.1 Descripción del modelo de dominio LRSVoronoi.....	25
2.4 Requisitos de software	25
2.4.1 Historias de usuario (HU)	25
2.4.2 Requisitos no funcionales (RNF)	28
2.5 Tarjetas Clase – Responsabilidad – Colaborador.....	28
2.6 Diagrama de clases	32
2.7 Patrones de diseño.....	33
2.7.1 Patrones para asignar responsabilidades (GRASP).....	33

Índice de contenido

Conclusiones del capítulo.....	35
CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA.....	36
3.1 Implementación.....	36
3.2 Pruebas de software.....	37
3.2.1 Pruebas de caja blanca	37
3.2.2 Pruebas de aceptación	42
3.3 Caso de estudio	44
Conclusiones del capítulo.....	48
Conclusiones generales	49
Recomendaciones.....	50
BIBLIOGRAFÍA	51

ÍNDICE DE FIGURAS

Figura 1. Diagrama de Voronoi.....	9
Figura 2. Regiones de Voronoi entre dos puntos.	10
Figura 3. Diagrama de Voronoi de un conjunto de puntos.	10
Figura 4. Propiedades del Diagrama de Voronoi.	11
Figura 5. Diagrama de Voronoi ponderado multiplicativo.....	12
Figura 6. Diagrama de Voronoi ponderado.....	24
Figura 7. Modelo de dominio del plugin LRSVoronoi.	25
Figura 8. Diagrama de clases de Servicio.	32
Figura 9. Diagrama de clases de LRSVoronoi.	32
Figura 10. Evidencia de patrón experto.	33
Figura 11. Evidencia del patrón creador.	34
Figura 12. Código del método VoronoiPonderado().....	39
Figura 13. Grafo de flujo.....	40
Figura 14. Regionalización del servicio de clínicas estomatológicas.	46
Figura 15. Regionalización del servicio de hospitales.....	47
Figura 16. Regionalización del servicio de policlínicos.	48

ÍNDICE DE TABLAS

Tabla 1. Ejemplo de la representación de las HU.	26
Tabla 2. HU-01.	27
Tabla 3. HU-03.	27
Tabla 4. Tarjeta CRC Voronoi.	28
Tabla 5. Tarjeta CRC ServicioDialog.	29
Tabla 6. Tarjeta CRC NuevoServicio.	29
Tabla 7. Tarjeta CRC VoronoiDialog.	30
Tabla 8. Tarjeta CRC NuevoServicioUi.	31
Tabla 9. Tarjeta CRC Servicio.	31
Tabla 10. Caso de prueba camino básico 1.	40
Tabla 11. Caso de prueba camino básico 2.	41
Tabla 12. Caso de prueba camino básico 3.	41
Tabla 13. Caso de prueba camino básico 4.	41
Tabla 14. Caso de prueba aceptación HU-01.	43
Tabla 15. Caso de prueba aceptación HU-02.	43
Tabla 16. Servicios de clínicas estomatológicas.	44
Tabla 17. Servicios de hospitales.	45
Tabla 18. Servicios de policlínicos.	45

INTRODUCCIÓN

En la actualidad los servicios son denominados como el sector terciario de la economía (ARRIAGADA 2007). Entre los principales servicios de este sector están: los de salud, educación, seguridad, administración de justicia, financieros estatales, de comercio exterior y de políticas económicas, por mencionar algunos. Se estima que entre el 70 y 80% del comercio mundial está en este sector (ALFONZO 2011), jugando un papel importante no sólo en la economía de algunos países, sino en la economía mundial.

Desde el diseño de los servicios se deben trazar las estrategias para garantizar su gestión, entendiéndose por gestión en el contexto de la investigación como: la planificación, aseguramiento de la calidad, localización y distribución (MURDICK *et al.* 1990). Al problema de la localización se le ha dado respuesta por medio de investigaciones procedentes de los más diversos campos, por ejemplo: urbanístico, análisis geográfico, política económica y administración de empresa (ILLERIS 1997; PARDOS *et al.* 2004). En el ámbito del análisis geográfico la solución más utilizada está asociada a los elementos de la matemática computacional (BRAVO 1999), basada en métricas de distancia euclidiana (MIGNONE 2013) y la teoría de grafo (CARDOZO *et al.* 2009), mediante algoritmos de caminos mínimos (SEGÚ PONS *et al.* 2003). La integración de estas soluciones a los Sistemas de Información Geográfica (SIG o GIS, en su acrónimo inglés) los han convertido en herramientas útiles para la solución al problema de la localización y selección del espacio (ESCOBAR MARTÍNEZ 1995).

Los SIG están dotados de funciones de gestión, almacenamiento, visualización y análisis de datos geográficos. También posibilitan la representación gráfica y la inserción de modelos para realizar análisis espacial (CAMPOS 1991). A estos se le han integrado soluciones de problemas específicos basados en diferentes enfoques como: algoritmos heurísticos y meta-heurísticos. Además se han propuesto soluciones basadas en la regionalización de los servicios, siendo la regionalización el proceso de delimitar el área a la que se le prestará el servicio. Para regionalización uno de los métodos más utilizados son los diagramas de Voronoi, que permiten dividir el espacio a partir de un conjunto predeterminado de puntos o generadores.

Teniendo en cuenta las potencialidades que brindan los diagramas de Voronoi en la regionalización del espacio, pueden ser utilizados para la distribución de los servicios. Esta distribución la realiza una unidad administrativa -de un cierto orden o rango- para garantizar

Introducción

que la población reciba el servicio existente en su región. En estos casos, el proceso de regionalización tiene como objetivo crear regiones equilibradas en lo que concierne a la asignación de recursos por habitante. Para realizar la regionalización de los servicios se deben tener en cuenta variables e indicadores, que garanticen la eficiencia y calidad del mismo. Los diagramas de Voronoi clásicos tienen limitantes a la hora de regionalizar los servicios, pues no integran las características que describen cada servicio(OKABE *et al.* 2009; SUGIHARA *et al.* 1992).

En Cuba dentro los lineamientos que guían el proceso de rectificación de la política económica, se hace énfasis en la necesidad de reorganizar, compactar y regionalizar los servicios de salud. En estas actividades es importante tener en cuenta las necesidades de cada provincia y municipio, de manera que cada paciente reciba la atención correspondiente con la calidad necesaria.

Se ha podido constatar que en Cuba los SIG en el sector de salud pública, se han utilizado principalmente en el análisis y distribución de los problemas de salud, enfocándose en la estadística médica o en los modelos epidemiológicos(FERNÁNDEZ NÚÑEZ 2006). No siendo así para la regionalización de los servicios, teniendo en cuenta indicadores de variada naturaleza.

Teniendo en cuenta lo antes expuesto, se plantea el siguiente **problema a resolver**: ¿cómo realizar la regionalización y localización de servicios de salud en Sistemas de Información Geográfica?

Para ello se define como **objeto de estudio**: la regionalización y localización de servicios.

Identificándose como **campo de acción**: los diagramas de Voronoi como método de regionalización.

Con el propósito de dar solución al problema planteado, se precisa como **objetivo general**: desarrollar una aplicación informática para la solución al problema de la regionalización y localización de servicios utilizando diagramas de Voronoi y, variables socioeconómicas y demográficas.

Para dar cumplimiento al objetivo general se desglosaron las siguientes **objetivos específicos**:

Introducción

1. Analizar los referentes teóricos en que se fundamenta la solución al problema de la regionalización y localización de servicios.
2. Diseñar una propuesta de integración de variables socioeconómicas y demográficas a los diagramas de Voronoi para su incorporación en la regionalización y localización de servicios.
3. Implementar una solución informática para la regionalización y localización de servicios utilizando diagramas de Voronoi y, variables socioeconómicas y demográficas.
4. Verificar la solución propuesta aplicando diferentes pruebas.

Idea a defender: el desarrollo de una aplicación informática para la solución al problema de la regionalización y localización de servicios utilizando diagramas de Voronoi y, variables socioeconómicas y demográficas, contribuirá al proceso de toma de decisiones en las instituciones de salud.

Posibles resultados: obtener un plugin para el Sistema de Información Geográfica Quantum GIS que permita la regionalización y localización de servicios utilizando los diagramas de Voronoi y, variables socioeconómicas y demográficas.

Utilizando para su validación: pruebas de aceptación y de caja blanca.

Para el desarrollo de la investigación se emplearon los siguientes métodos científicos:

Los **métodos teóricos** posibilitan las condiciones para buscar más que las características triviales de la realidad, permiten explicar los hechos y profundizar en las principales relaciones y cualidades de los fenómenos, hechos y procesos.

- **Modelación:** se utiliza para la confección de los diagramas correspondientes a la fase de análisis y diseño convirtiéndose en una guía para la implementación de la solución propuesta.
- **Analítico- Sintético:** empleado para analizar la bibliografía correspondiente a la investigación y extraer los elementos significativos para el diseño e implementación de la herramienta.
- **Hipotético - Deductivo:** para el análisis y la definición de la idea a defender que será verificada o probada con el desarrollo de la investigación, arribando a conclusiones particulares.

Introducción

Los **métodos empíricos** describen y explican las características del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia.

- **Entrevista:** esta técnica se utilizó con el objetivo de conocer los elementos necesarios para la implementación de la aplicación.

Para lograr la claridad y comprensión de los contenidos de la investigación realizada se ha estructurado el documento en tres capítulos.

En el **capítulo 1** se realiza el estudio del estado del arte, se hace énfasis en los conceptos asociados a la investigación así como las soluciones existentes a nivel mundial. Además se caracteriza la metodología de desarrollo de software, se detallan las principales tecnologías y herramientas que se van a utilizar en la construcción de la propuesta de solución, así como las ventajas de su utilización.

En el **capítulo 2** se presenta la propuesta de solución basada en diagramas de Voronoi ponderados. Se especifican los patrones del diseño aplicados y los artefactos derivados de la metodología de desarrollo de software seleccionada.

En el **capítulo 3** se abordan todos los elementos relacionados con la implementación, se elaboran y documentan las pruebas para verificar el correcto funcionamiento de la solución propuesta. Finalmente se presentan los resultados obtenidos a partir de la aplicación a un caso de estudio.

Capítulo 1. El problema de la localización y regionalización de servicios

CAPÍTULO 1. EL PROBLEMA DE LA LOCALIZACIÓN Y REGIONALIZACIÓN DE SERVICIOS

El presente capítulo expone los fundamentos teórico-prácticos relacionado con el problema planteado. Se analizan algunos SIG, con el fin obtener conocimiento sobre cómo se trata la localización y la regionalización de servicios en estos sistemas, además se caracterizan las herramientas a utilizar durante el proceso de desarrollo de la propuesta de solución.

1.1 Conceptos asociados al dominio del problema

Localización y regionalización de servicios

En el contexto de esta investigación se trata la localización como la ubicación geográfica de uno o más servicios, para atender a un conjunto determinado de personas. Entiéndase por servicio: un hospital, un policlínico o consultorio de la familia. Por otra parte la regionalización consiste en la delimitación geográfica de dichos servicios, una vez efectuada su localización, mediante indicadores asociados a las variables socioeconómicas y demográficas.

Cartografía

“La cartografía es la ciencia que se encarga del estudio y elaboración de los mapas geográficos, territoriales y de diferentes dimensiones lineales. Consiste en la aplicación de métodos, técnicas, instrumentos y conocimiento para el diseño de cartografía básica, cartografía digital y sistemas de información territorial; constituye un instrumento fundamental para la comunicación de información espacial y la toma de decisiones”(RAISZ and MANTERO 1985). Se entiende por cartografía a la ciencia encargada de realizar el estudio y trazado de los mapas, con el objetivo de obtener información útil de ellos.

Sistemas de Información Geográfica

Un SIG es básicamente, un “sistema de hardware, software y procedimientos, diseñado para realizar la captura, almacenamiento, manipulación, análisis, modelización y presentación de datos referenciados espacialmente, para la resolución de problemas complejos de planificación y gestión”(PUEBLA and GOULD 1994).

Capítulo 1. El problema de la localización y regionalización de servicios

Análisis exploratorio de datos espaciales

El análisis de los datos espaciales es una disciplina dentro del análisis estadístico más general, diseñada para el tratamiento específico de los datos geográficos. El Análisis exploratorio de datos espaciales se utiliza para identificar relaciones sistemáticas entre variables, o dentro de una misma variable, cuando no existe un conocimiento claro sobre su distribución en el espacio geográfico(CHASCO YRIGOYEN 2002).

Datos Espaciales

“Los Datos Espaciales son el componente fundamental de cada proyecto o aplicación SIG. Contienen las ubicaciones y formas de características cartográficas. Son también conocidos como Datos Cartográficos Digitales, el tipo de datos necesarios para crear mapas y estudiar relaciones espaciales. Dentro de su contexto, almacenan información sobre la localización, las formas de un objeto geográfico y las relaciones entre ellos, normalmente con coordenadas y topología”(I SUBIRANA 2004).

Análisis Espacial

Se define en “Geographic Information Systems and Science” como Análisis Espacial “...todas las transformaciones, la administración, y los métodos que pueden ser aplicados a los datos geográficos para agregarles valor para el soporte de decisiones, mostrar y revelar patrones y anomalías que no son inmediatamente obvias”(LONGLEY *et al.* 2001). En otras palabras, el análisis espacial es el proceso en el cual se convierten los datos iniciales en información útil.

Interpolación espacial

La interpolación espacial es un procedimiento matemático utilizado para predecir el valor de un atributo en una localidad específica a partir de valores obtenidos de puntos vecinos a él(MIRANDA-SALAS and CONDAL 2003).

Entre los disímiles métodos de interpolación espacial se encuentran:

- Método de la distancia inversa (IDW).
- Spline Cúbico.
- Interpolación lineal.
- Diagramas de Voronoi.

Capítulo 1. El problema de la localización y regionalización de servicios

1.2 Caracterización del análisis espacial basado en métodos de interpolación

Uno de los objetivos que persigue el análisis espacial de los datos es obtener información temática cartográfica, representada a través de mapas que reflejen de la forma más precisa el área de estudio.

Se coincide con (CHICA OLMO *et al.* 1995) en dos cuestiones que llaman la atención debido a su importancia en el estudio del análisis espacial de los datos:

1. Los datos: aquella parte de la realidad desconocida a la cual podemos acceder experimentalmente.
2. Los métodos de análisis numéricos: se aplican al estudio de los datos obtenidos.

En esencia el análisis espacial de los datos posibilita identificar los componentes del espacio analizado (datos); luego se realiza un procedimiento o conjunto de estos (métodos de análisis numérico) que permitan comprender la funcionalidad de los componentes obtenidos. Es por ello que el resultado alcanzado depende de la selección del método utilizado.

El empleo de métodos de análisis numérico provocado por el creciente avance de las TIC ha posibilitado que las ciencias matemáticas se desarrollen favorecidas por la computación. La interpolación espacial es uno de los procesos que se desarrolla sobre bases matemáticas, lo que posibilita exactitud en los resultados.

Cuando se refiere a interpolación espacial se puede definir como la estimación de los valores que alcanza una variable Z en un conjunto de puntos definidos por un par de coordenadas (X, Y). Los SIG emplean la interpolación espacial para obtener capas que constituyen la variable a interpolar. Por lo que cada celda en la capa se convierte en un punto a interpolar.

Los métodos de interpolación se basan en la semejanza que toman los valores de cualquier variable cuantitativa que se mida entre dos puntos, mientras más cercanos estén estos. No se puede afirmar que un método es superior a otro, ya que cada uno cumple una función determinada. El empleo de un método en específico está en dependencia de la situación espacial existente.

Descripción de proceso de interpolación

Para la realización de una correcta interpolación espacial se debe primeramente construir la grilla o malla que permite la regularización de los valores de Z, de forma que la estimación para el punto de la grilla sea lo más cercano posible al valor esperado de Z.

Capítulo 1. El problema de la localización y regionalización de servicios

Al construir la malla se forma una cuadrícula, que se realiza mediante la utilización de un método de interpolación, en el caso de esta investigación se realiza mediante los diagramas de Voronoi.

Cuando los valores están regularizados se realiza la interpolación aplicando un método que puede ser el mismo que se empleó para construir la grilla u otro de los mencionados anteriormente. Este método utiliza funciones matemáticas que permiten establecer la estimación de los valores que alcanza una variable Z (valores a interpolar) en un conjunto de puntos definidos por un par de coordenadas (X, Y); obteniendo como resultado final una función que pase lo más cerca posible por los valores resultantes del proceso. Una vez formada la función que recorre todos los puntos se realiza el suavizado de contornos para mejorar la visualización de los resultados(MIRANDA-SALAS and CONDAL 2003).

1.3 Caracterización de los métodos de interpolación

Método de la distancia inversa (IDW)

Se basa en la auto-correlación espacial de los puntos para la predicción y generación de superficies continuas, por lo que, parte de la teoría que las cosas que están más cerca son más parecidas y por tanto tienen más peso e influencia sobre el punto a estimar(MURILLO *et al.* 2012).

“La ventaja de IDW es que es intuitiva y eficiente. Esta interpolación funciona mejor con puntos distribuidos uniformemente”(ANDERSON 2002).

Spline Cúbico

Estima valores analizados usando una función matemática que minimiza la curvatura general de la superficie, dando como resultado una superficie lisa que pasa exactamente a través de los puntos de entrada. Este método es el mejor para suavizar superficies distintas, como la elevación, la altura de la capa freática, o las concentraciones de contaminación(KAWAMURA *et al.* 1992).

Ofrece un resultado continuo, suave y adaptable, si se ajustan adecuadamente sus parámetros se suele llegar a una interpolación de mayor precisión(PESQUER *et al.* 2007).

Capítulo 1. El problema de la localización y regionalización de servicios

Interpolación lineal

La interpolación lineal es un método relativamente sencillo que se basa en la suposición de que la tasa de cambio entre los valores conocidos es constante y puede calcularse a partir de estos valores utilizando la ecuación de la pendiente simple. Entonces, un valor desconocido entre los dos puntos conocidos se puede calcular usando uno de los puntos y la tasa de cambio. Proporciona resultados de forma rápida y sencilla, debido a que solo realiza el cálculo de trayectoria de dos puntos (MATHEWS *et al.* 2000).

Diagramas Voronoi

Un diagrama de Voronoi (DV), mencionado en primer lugar por Georges Voronoi (VORONOI 1908), es una manera de dividir el espacio en varias regiones de un conjunto de puntos (llamados semillas, sitios o generadores) como se muestra en la Figura 1. Para cada generador habrá una región correspondiente (llamado una región Voronoi o una celda de Voronoi) que consta de todos los puntos más cercana a la semilla que a cualquier otra.

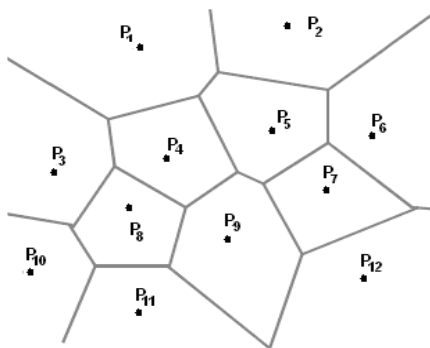


Figura 1. Diagrama de Voronoi.

El diagrama se crea al unir los puntos entre sí, trazando las mediatrices de los segmentos de unión. Las intersecciones de estas mediatrices determinan una serie de polígonos en un espacio bidimensional alrededor de un conjunto de puntos de control, de manera que el perímetro de los polígonos generados sea equidistante a los puntos vecinos y designando su área de influencia.

Dados 2 puntos, P_i y P_j en un plano T , la perpendicular al segmento P_iP_j en su punto medio divide el plano en dos regiones V_i y V_j ; la región V_i contiene todos y solo los puntos cuya distancia a P_i es menor que a P_j y la región V_j contiene el resto.

Capítulo 1. El problema de la localización y regionalización de servicios

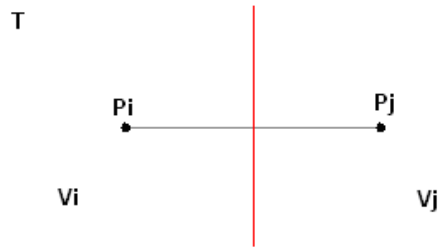


Figura 2. Regiones de Voronoi entre dos puntos.

El lugar geométrico de los puntos que están más cerca de un punto P_i que del resto de puntos se denomina **región de Voronoi del punto P_i** , V_i , y al punto P_i su punto generador.

El concepto se extiende a múltiples puntos P_n de forma que cada uno se le asocia a una región de Voronoi V_n , que contiene todos los puntos del plano más próximos. Si se aplica a un dominio cerrado se genera un conjunto de polígonos convexos que divide el plano.

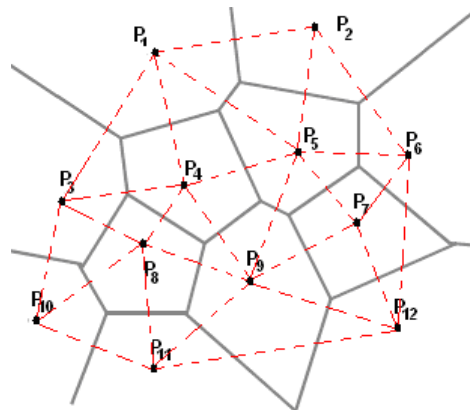


Figura 3. Diagrama de Voronoi de un conjunto de puntos.

Al conjunto de puntos p_1, \dots, p_n se denomina **conjunto generador** del Diagrama de Voronoi.

Algunas propiedades del diagrama de Voronoi:

1. Dos puntos P_i y P_j son vecinos si comparten una arista. Una arista es la bisectriz perpendicular del segmento $P_i P_j$.
2. Un vértice de Voronoi es un punto equidistante a tres generadores (si lo es a más de tres hablamos de casos degenerados) y es la intersección de tres aristas.
3. Dentro del círculo con centro en un vértice de Voronoi y que pasa por 3 puntos generadores no puede existir ningún otro punto generador.

Capítulo 1. El problema de la localización y regionalización de servicios

4. Una región de Voronoi es no acotada si su punto generador pertenece a la envolvente convexa de la nube de puntos.
5. Una región de Voronoi es un polígono convexo o es una región no acotada.
6. El diagrama de Voronoi de un conjunto de puntos es único (RODRÍGUEZ GAVILÁN 2009).

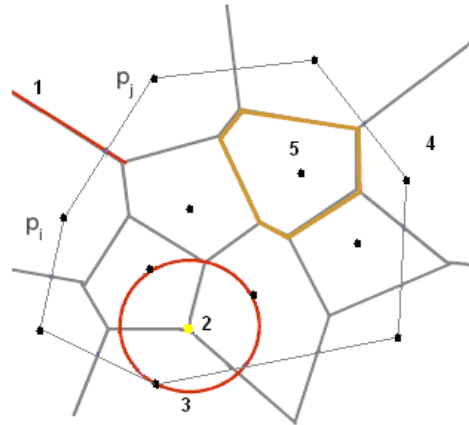


Figura 4. Propiedades del Diagrama de Voronoi.

Teniendo en cuenta estas propiedades de los diagrama de Voronoi se puede concluir que un punto x del plano pertenece al área de influencia del punto generador p_i , denominada polígono de Voronoi, V_i , si y solo si el punto x está más cerca de p_i que cualquier otro punto generador (DE BERG *et al.* 2000).

El clásico diagrama de Voronoi presenta los puntos generadores con un mismo peso y su construcción depende solamente de la distancia que existe entre estos puntos. En el presente trabajo se optó por utilizar un método generalizado de estos diagramas, designado como diagrama de Voronoi ponderado.

Diagrama de Voronoi ponderado

Un diagrama de Voronoi ponderado (DVP) es un diagrama de Voronoi para los que se definen los polígonos de Voronoi, en términos de una distancia definida por algunos sistemas de medición comunes, modificados por pesos asignados al punto generador (OKABE *et al.* 2000). El tipo de diagrama de Voronoi ponderado que se analizará es el multiplicativo (DVPM), y se define cuando la distancia entre los puntos se multiplica por pesos positivos (DEZA and DEZA 2006). En el plano, bajo la distancia euclidiana ordinaria, el DVPM también se llama una teselación Dirichlet circular (ASH and BOLKER 1986) y sus bordes son arcos circulares o segmentos de línea recta.

Capítulo 1. El problema de la localización y regionalización de servicios

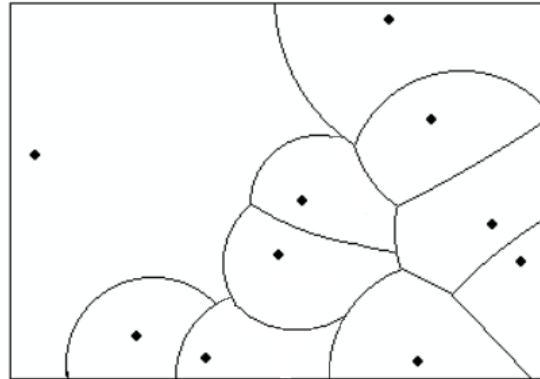


Figura 5. Diagrama de Voronoi ponderado multiplicativo.

1.4 Análisis de las soluciones existentes

En este epígrafe se analizan algunos SIG utilizados en la actualidad para la localización de servicios. Así como sistemas y propuestas de solución para la regionalización de espacios utilizando diagramas de Voronoi. Prestando especial atención a la forma en que dan solución al problema de la localización y qué variante de los diagramas de Voronoi utilizan para la regionalización o asignación de espacios.

OsmAnd y Andariego

Dentro de los SIG estudiados se encuentran el OsmAnd y Andariego, que están desarrollados para telefonía móvil y son similares con respecto a las características que presentan. Ambos realizan localizaciones puntuales de servicios utilizando siempre la distancia euclidiana, en función del camino mínimo y teniendo en cuenta las rutas viales. No presentan ninguna funcionalidad que permita la regionalización de espacios.

RATM (Relocalización de antenas para la mejora de una red de telefonía móvil)

Este sistema fue desarrollado en la Universidad Politécnica de Madrid y su función es relocalizar las antenas de telefonía móvil, teniendo en cuenta qué región abarca cada una. El autor de este trabajo se basa en el algoritmo de Voronoi, en conjunto con un algoritmo genético. Una vez generado el diagrama de Voronoi, el algoritmo genético calcula el área de la región de Voronoi y verifica si la misma está en correspondencia con el rango que puede abarcar la antena. Esta operación se realiza teniendo en cuenta una serie de indicadores definidos por el autor.

Capítulo 1. El problema de la localización y regionalización de servicios

Quantum GIS

Dentro de los SIG libres más usados se encuentra el QGIS, desarrollado por la Fundación OSGeo. Este permite una gran cantidad de plugins que aportan los complementos necesarios para desarrollar complejos análisis espaciales, así como generación de mapas, estadísticas, publicación online y conexión con bases de datos espaciales (ALONSO 2015). Además cuenta con una funcionalidad de representar los diagramas de Voronoi clásico a partir de un conjunto de puntos.

Algoritmo MWVD (Multiplicatively Weighted Voronoi Diagrams)

Este es un algoritmo propuesto por (SONG *et al.* 2014) para generar los diagramas de Voronoi. El algoritmo se basa en la definición de diagrama de Voronoi ponderado de (OKABE *et al.* 2000) y garantiza una representación de los diagramas de Voronoi en correspondencia con los pesos de cada punto generador.

Conclusiones del epígrafe

Después de analizados los sistemas y propuestas de soluciones para tratar los problemas de la localización y regionalización de espacios se puede concluir que:

- Los SIG analizados no cumplen con todas las características deseadas, sin embargo cuentan con funcionalidades que se pueden tener en cuenta para el desarrollo de una solución.
- Las soluciones más cercanas a las características analizadas, están diseñadas para tratar problemas específicos que no pueden generalizarse para resolver otros problemas.

1.5 Metodología de desarrollo de software

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

Capítulo 1. El problema de la localización y regionalización de servicios

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

Programación Extrema (XP)

La metodología XP está centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. Esta se basa en realimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las soluciones implementadas y coraje para enfrentar los cambios (BECK 2000).

Características de XP (BECK 2000):

- XP es una metodología “liviana” que no tiene en cuenta la utilización de elaborados casos de uso y la generación de una extensa documentación.
- XP tiene asociado un ciclo de vida y es considerado a su vez un proceso.
- La tendencia de entregar software en espacios de tiempo cada vez más pequeños con exigencias de costos reducidos y altos estándares de calidad.
- XP define Historias de Usuario como base del software a desarrollar, estas historias las escribe el cliente y describen escenarios sobre el funcionamiento del programa. A partir de las historias de usuario 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 historias de usuario.
- **Diseño:** Durante esta etapa se crea un diseño evolutivo que va mejorando incrementalmente y que permite hacer entregas pequeñas y frecuentes de valor para

Capítulo 1. El problema de la localización y regionalización de servicios

el cliente, basado principalmente en el desarrollo de las tarjetas Clase-Responsabilidad- Colaboración (CRC).

- **Desarrollo:** En esta fase se realiza la implementación de las Historias de usuario (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.

1.6 Herramientas y lenguajes

En este epígrafe se realiza un estudio de las herramientas y lenguajes a utilizar en el desarrollo de la aplicación. De las mismas se analizan sus características y funcionalidades, con el objetivo de incorporarlas a la solución del problema planteado.

1.6.1 Lenguaje Unificado de Modelado (UML) 2.0

UML es un lenguaje que permite modelar lo que el software deberá hacer; posibilitando un mayor entendimiento y comprensión del sistema por parte del equipo de desarrollo. Sus objetivos son varios, pero se pueden resumir en sus funciones(BOOCH *et al.* 1999):

- **Visualizar:** Posibilita expresar de una forma gráfica un sistema, de manera que pueda ser entendido fácilmente.
- **Especificar:** UML permite especificar cuáles son las características de un sistema antes de su construcción.
- **Construir:** A partir de los modelos especificados se pueden construir los sistemas diseñados.
- **Documentar:** Los propios elementos gráficos sirven como documentación del sistema desarrollado, los que pueden ser empleados en su futura revisión.

Capítulo 1. El problema de la localización y regionalización de servicios

1.6.2 Herramientas CASE

Las herramientas CASE¹ actualmente brindan una gran cantidad de componentes que incluyen la mayoría de los requisitos necesarios para el desarrollo de los sistemas, han sido creadas con una gran exactitud en cuanto a las necesidades de los desarrolladores de software para la automatización de procesos incluyendo el análisis, diseño e implantación.

En la actualidad muchas empresas se han extendido a la adquisición de herramientas CASE, con el fin de automatizar sus procedimientos administrativos. Dentro de estas herramientas se encuentra Visual Paradigm for UML.

1.6.3 Visual Paradigm for UML Enterprise Edition 8.0

Es una herramienta CASE para desarrollo de aplicaciones utilizando modelado UML. Posibilita guiar todo el proceso de desarrollo del software, desde la fase de análisis hasta la realización de las pruebas garantizando reducir al máximo los costes y posibilitando la creación de un sistema de alta calidad.

Las principales características de la herramienta son:

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

Visual Paradigm 8.0 facilitó la creación de los diagramas que describen el ciclo de vida del sistema a través de un diseño centrado en casos de uso y enfocado al negocio, garantizando realizar un software con mayor calidad. Es fácil de instalar y actualizar, está disponible para software libre y tiene licencia gratuita. Luego de modelar el sistema se realiza la programación del mismo utilizando los lenguajes JavaScript 1.5, python 2.7 y Preprocesador de Hipertexto 5 (PARADIGM 2010).

¹CASE: siglas en inglés que se utilizan para referirse a Ingeniería de Software Asistida por Computadora

Capítulo 1. El problema de la localización y regionalización de servicios

1.6.4 Lenguajes de programación

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

Para el desarrollo de la aplicación es necesario utilizar un lenguaje que facilite el trabajo con el sistema QGIS, teniendo en cuenta las características que este presentase utiliza como lenguaje de programación Python.

Python 2.7

Python es un lenguaje orientado a objetos e interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, esto ofrece ventajas como la rapidez de desarrollo. Integra gran cantidad de bibliotecas que contienen tipos de datos y funciones incorporadas en el propio lenguaje, permitiendo realizar muchas tareas sin necesidad de tener que programarlas desde cero (DUQUE 2011).

Es importante mencionar que el sistema QGIS trae soporte para este lenguaje a partir de su versión 0.9 y en conjunto con la librería PyQt permite el desarrollo de interfaces gráficas de usuario.

PyQt 4

PyQt es un conjunto de enlaces Python para el conjunto de herramientas Qt y está disponible para todas las plataformas soportadas por Qt, incluyendo Windows, Linux, UNIX, Mac OS / X, entre otros. Para elaborar prototipos de alta calidad se utiliza QtDesigner, pues PyQt lo distingue por su sencillez de poseer herramientas que gestionen su manipulación, por ser adaptable a distintas plataformas de software (SUMMERFIELD 2007).

QtDesigner 4.7.4

QtDesigner es la herramienta Qt para el diseño y la creación de interfaces gráficas de usuario. Este permite diseñar ventanas completas mediante formularios y una interfaz de arrastrar y soltar. Tiene la posibilidad de pre-visualizar sus diseños para asegurarse de que funcionan como usted pretende, y que le permite crear prototipos con sus usuarios, antes de tener que escribir ningún código (LIMITED 2014).

Capítulo 1. El problema de la localización y regionalización de servicios

1.6.5 Entorno de Desarrollo Integrado (IDE)

Un entorno de desarrollo integrado (IDE, por sus siglas en inglés) es una herramienta que permite a los desarrolladores de software escribir sus programas en uno o más lenguajes. Consiste básicamente en una plataforma en la que se integran un editor de código, un compilador, un depurador y una interfaz gráfica de usuario (KERRIGAN *et al.* 2007).

PyCharm 3.4

Para el trabajo con las bibliotecas del lenguaje python se utilizó como IDE PyCharm por las facilidades de trabajo que brinda, además de su fuerte integración y organización con python. En (JETBRAINS 2014) se plantea que “PyCharm es un IDE para programar en el lenguaje python que contiene un editor inteligente, posibilitando el completamiento de código, asistencia y análisis del mismo, así como herramientas de depuración y prueba”.

1.6.6 Sistema de Información Geográfica Quantum GIS (QGIS) 2.6

QGIS es “un sistema de información geográfica que permite visualizar, editar e imprimir mapas, así como analizar todo tipo de datos demográficos y económicos usando la geografía como referencia. La variedad y potencia de las herramientas, así como la gran cantidad de documentación y ejemplos disponibles, hacen de Quantum GIS una de las mejores utilidades de análisis de datos geográficos disponible” (TEAM 2012). Además es programa de código abierto y gratuito que permite la incorporación de nuevos complementos o plugins.

Plugin

Un plugin es una aplicación que incrementa una característica o servicio a un sistema ya existente; y éste se ejecuta por la aplicación principal. A nivel mundial, existen aplicaciones basadas en arquitectura plugin que le brindan la capacidad de agregar funcionalidades nuevas en tiempo de ejecución. Entre sus principales ventajas está que permite que los desarrolladores externos colaboren con la aplicación, de modo que se extiendan sus funcionalidades, posibilita la personalización de la aplicación de manera no pensada por el autor, o que al menos no tenía la intención de hacer. Los plugins en el QGIS tienen gran importancia, estos le brindan al mismo un gran potencial para la adaptación y el desarrollo de nuevas funcionalidades.

Capítulo 1. El problema de la localización y regionalización de servicios

Conclusiones del capítulo

A partir del análisis del estado del arte se puede concluir que los diagramas de Voronoi clásicos presentan limitaciones para regionalizar territorios teniendo en cuenta indicadores diferentes a la distancia entre dos puntos, estas limitantes pueden ser tratadas mediante los diagramas de Voronoi ponderados, pues asocia a la construcción de una región un peso asociado a cada punto. El análisis de las soluciones existentes, evidenció las carencias que presentan para regionalizar el espacio de forma eficiente incluyendo variables de diversa naturaleza. El SIG QGIS es una alternativa libre para la solución a problemas de análisis espacial sin embargo no cuenta con una funcionalidad que le permita regionalizar los servicios incluyendo variables que pueden incidir sobre su calidad.

Capítulo 2. Descripción de la propuesta de solución

CAPÍTULO 2. DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN

En este capítulo se presenta la propuesta de solución, las características que tendrá la misma y se presentan los principales artefactos generados según la metodología XP.

2.1 Propuesta de solución

Para una mejor integración de los DV y DVPM con SIG para regionalizar los servicios, se propone un algoritmo basado en vectores para generar DV y DVPM tomando como referencia el trabajo de (SONG *et al.* 2014), de ahí que:

Sea $P = \{p_1, p_2, \dots, p_n\}$ un conjunto de n puntos distintos en \mathbb{R}^2 de coordenadas $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$. El subdivide \mathbb{R}^2 en n regiones de Voronoi $V(p_i)$, con la propiedad de que un punto $q(x, y)$ se encuentra en la región de Voronoi $V(p_i)$ si y sólo si $d(p_i, q) < d(p_j, q)$ para cada $p_i, p_j \in P$ con $i \neq j$, se define como el diagrama de Voronoi $V(P) = \{V(p_1), V(p_2), \dots, V(p_n)\}$, brevemente DV. La denotación $d(p_i, q)$ denota una función determinada distancia entre el sitio de p_i y el punto q . Un $DV_{Vo}(P)$ es un diagrama de Voronoi usando la métrica euclidiana, definido por:

$$d_c(p_i, q) = \|p_i - q\| = \sqrt{(x_i - x)^2 + (y_i - y)^2}$$

Sea $P = \{p_1, p_2, \dots, p_n\}$ un conjunto de n puntos distintos en \mathbb{R}^2 , $W = \{w_1, w_2, \dots, w_n\}$ un conjunto de n números reales positivos. A cada $p_i \in P$ le corresponde a un parámetro $w_i \in W$. Al subdividir \mathbb{R}^2 en n regiones de Voronoi multiplicativas $V_{pm}(p_i)$, con la propiedad de que un punto $q(x, y)$ se encuentra en la región $V_{pm}(p_i)$ si y sólo si $d(p_i, q) / w_i < d(p_j, q) / w_j$ para cada $p_i, p_j \in P$ con $i \neq j$, se define como el diagrama de Voronoi multiplicativo $V_{pm}(P, W) = \{V_{pm}(p_1, w_1), V_{pm}(p_2, w_2), \dots, V_{pm}(p_n, w_n)\}$, brevemente DVPM. Los DVPM utilizan la siguiente función distancia entre un generador p_i con su peso asignado $w_i \in W$ y un punto de $q(x, y)$:

$$d_c(p_i, q) = \frac{\|p_i - q\|}{w_i} = \frac{\sqrt{(x_i - x)^2 + (y_i - y)^2}}{w_i}$$

Sea $p_i(x_i, y_i), p_j(x_j, y_j)$ dos puntos con pesos w_i y w_j , L_{ij} la bisectriz de p_i y p_j , L_{ij} de un DV se expresa como:

$$\sqrt{(x - x_i)^2 + (y - y_i)^2} = \sqrt{(x - x_j)^2 + (y - y_j)^2}$$

Capítulo 2. Descripción de la propuesta de solución

Después de la reducción, se puede obtener:

$$2(x_i - x_j)x + 2(y_i - y_j)y - (x_i^2 + y_i^2 - x_j^2 - y_j^2) = 0$$

Claramente, L_{ij} es una bisectriz perpendicular que separa todos los puntos en \mathbb{R}^2 más cerca de p_i de aquellos más cerca de p_j . Para una DVPM, L_{ij} de p_i y p_j pueden ser representados como:

$$\frac{\sqrt{(x - x_i)^2 + (y - y_i)^2}}{w_i} = \frac{\sqrt{(x - x_j)^2 + (y - y_j)^2}}{w_j}$$

Después de despejar y simplificar, podemos obtener:

$$\left(x - \frac{w_i^2 x_j - w_j^2 x_i}{w_i^2 - w_j^2}\right)^2 + \left(y - \frac{w_i^2 y_j - w_j^2 y_i}{w_i^2 - w_j^2}\right)^2 = \left(\frac{w_i w_j}{|w_i^2 - w_j^2|} d(p_i, p_j)\right)^2$$

En este caso $d(p_i, p_j)$ es la distancia euclidiana entre p_i y p_j .

Se sabe que la bisectriz de p_i y p_j forma un arco circular y la coordenada del punto central del círculo es:

$$\left(\frac{w_i^2 x_j - w_j^2 x_i}{w_i^2 - w_j^2}, \frac{w_i^2 y_j - w_j^2 y_i}{w_i^2 - w_j^2}\right)$$

Y para el radio:

$$\frac{w_i w_j}{|w_i^2 - w_j^2|} d(p_i, p_j)$$

al igual que en (SONG *et al.* 2014).

Para la obtención de los pesos de cada servicio, se debe utilizar una función que tenga en cuenta los indicadores a partir del problema que se va a resolver. En el caso de esta investigación se presenta un caso de estudio donde se definieron los indicadores a tener en cuenta y que función se utiliza para el cálculo del peso.

Capítulo 2. Descripción de la propuesta de solución

Pseudocódigo del algoritmo VoronoiPonderado

Explicada la propuesta solución, se presenta el pseudocódigo del algoritmo VoronoiPonderado.

Algoritmo VoronoiPonderado

Entrada:

- Capa de servicios (**capa**)
- Tipo de servicio (**tiposervicio**)

Salida

- Conjunto de n regiones que responden al polígono asociado a cada servicio (**V []**)

Inicio

1. Inicializar una lista de servicios (**puntos**)
 2. Inicializar una lista para guardar los pesos (**w**)
 3. Inicializar la lista para guardar las regiones de cada servicio (**V**)
 4. delimitar la región que abarcan todos los servicios (**drv**)
 5. puntosRegionalizar (**capa, puntos, drv, tiposervicio**)
 6. calcularPeso (**puntos, w**)
 7. Agregar a **V** la región **drv**
 8. Para toda **j** entre 0 y final -1 de **puntos**
 9. **Pj** = valor de **puntos** en **j**
 10. Guardar el valor de **V[j]** en una variable **polig** y eliminarlo de **V**
 11. **wj** = peso de **Pj**
 12. Inicializar una lista **aux**
 13. Para toda **i** entre **j+1** y final de **puntos**
 14. **Pi** = valor de **puntos** en **i**
 15. **wi** = peso de **Pi**
 16. Si **wi = wj**
 17. **geo** = divideBisectriz (**wj, Pi, Pj, polig**)
 18. Sino
 19. **geo** = divideRegionArc (**Pi, Pj, wi, wj, polig**)
 20. Añadir a **aux** el valor de **geo**
 21. **inter** = unión entre los polígonos de **aux**
 22. Insertar en **V** la diferencia entre **polig** e **inter**
 23. Añadir **inter** a **V**
 24. Devolver **V**
 25. Fin
-

Capítulo 2. Descripción de la propuesta de solución

En el pseudocódigo aparecen cuatro nuevos procedimientos:

- **puntosRegionalizar()**
- **calcularPeso()**
- **divideBisectriz()**
- **divideRegionArc()**

El procedimiento **puntosRegionalizar()** consiste en llenar la lista **puntos** pasada por parámetro y la lista **drv** filtrando los datos que contiene el parámetro **capa** según **tiposervicio**. Por otra parte el procedimiento **calcularPeso()** calcula el peso teniendo en cuenta el **servicio** especificado, mediante la función definida para el cálculo de peso.

Los procedimientos **divideBisectriz()** y **divideRegionArc()**, son los encargados de dividir una región $V[i]$ en dos regiones: $V[i]$ y $V[j]$, obteniendo como salida $V[j]$. La diferencia entre estos procedimientos es que, el primero divide la región con una bisectriz perpendicular al segmento que crean los puntos P_i y P_j y el segundo lo hace con un arco de circunferencia cerrando siempre para el punto de menor peso (w).

2.2 Complejidad del algoritmo

En este epígrafe se calcula la complejidad temporal, según el enfoque teórico, del algoritmo. Para el cálculo de la misma se determina la complejidad de cada paso del algoritmo y se utilizan las reglas de la suma y la multiplicación de la notación asintótica “O grande” (O) para obtener la complejidad de dicho algoritmo.

A continuación se explica la complejidad del algoritmo en cada paso:

Paso 1 $O(n)$

En este paso se realiza una llamada al método **puntosExtremos()** implementado, el cual tiene una complejidad $O(n)$, debido a que recorre una lista f de longitud n con complejidad $O(n)$ y las demás instrucciones dentro del método son operaciones primitivas sobre f que tienen una complejidad de $O(1)$ y al aplicar la regla de la suma se puede deducir la complejidad anteriormente afirmada.

Paso 2 $O(1)$

Las operaciones primitivas sobre la lista V toman tiempo constante, de ahí que la complejidad sea $O(1)$.

Capítulo 2. Descripción de la propuesta de solución

Paso 3 $O(n^2)$

En este paso se encuentra un ciclo anidado sobre una lista P de complejidad $O(n^2)$ y se le realizan llamadas a los métodos *radio ()*, *coordenadasArc ()* que realizan funciones primitivas de complejidad $O(1)$. Aplicando las reglas de la suma y multiplicación se puede deducir la complejidad del algoritmo es de $O(n^2)$.

Al aplicar esta propuesta en un entorno real debe obtenerse algo similar a lo que se representa en la figura 6, tomando los puntos de color marrón como puntos de servicios.

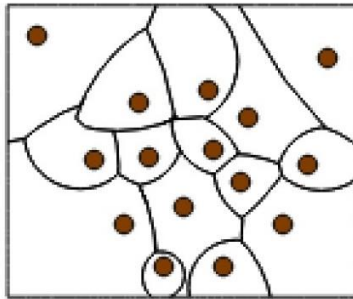


Figura 6. Diagrama de Voronoi ponderado.

2.3 Modelo de dominio

El modelo del dominio del problema define un modelo de clases común para todos los involucrados en el modelo de requisitos, analistas al igual que clientes. Las clases consisten de los objetos del dominio del problema, o sea objetos que tienen una correspondencia directa en el área de la aplicación. Como los usuarios y clientes deberían reconocer todos los conceptos, se puede desarrollar una terminología común al razonar sobre los casos de uso, y por lo tanto disminuyendo la probabilidad de malos entendimientos entre el analista y el usuario. Al discutirlo, se evolucionará el modelo del dominio del problema. Una técnica utilizada cuando se trabaja con tal modelo es darle al cliente un papel y un lápiz y pedirle que dibuje su visión del sistema (BRUNO 2014).

Capítulo 2. Descripción de la propuesta de solución

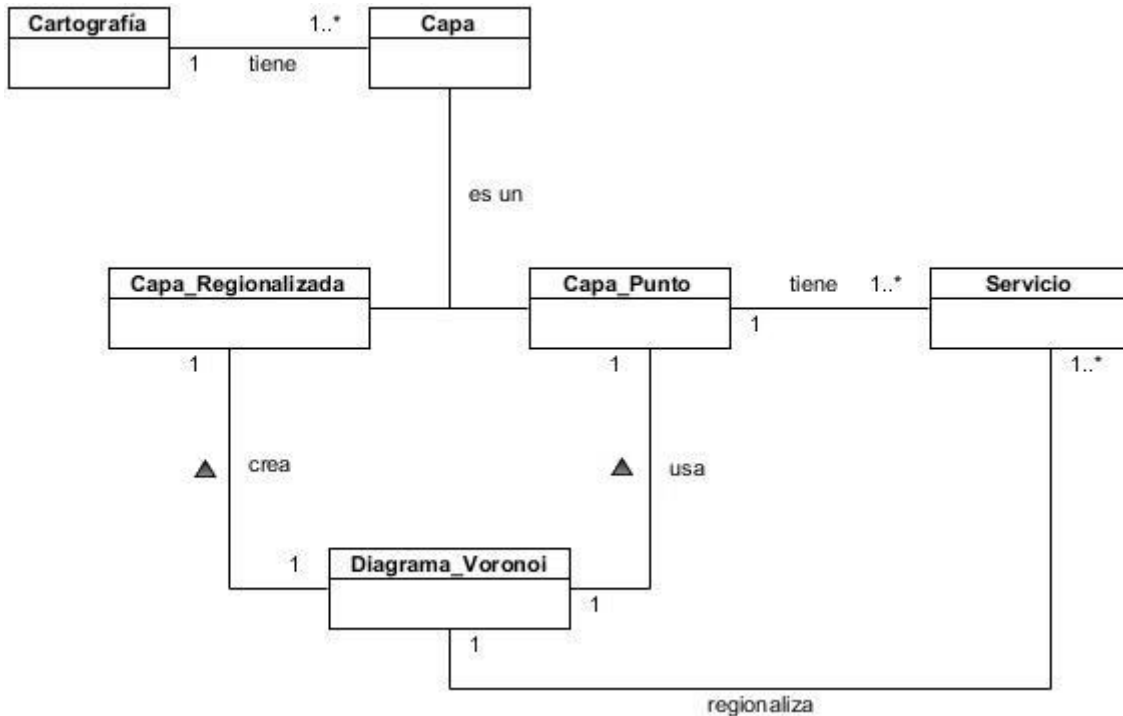


Figura 7. Modelo de dominio del plugin LRSVoronoi.

2.3.1 Descripción del modelo de dominio LRSVoronoi

El modelo de dominio es una visualización de elementos que están presente en la solución de nuestro problema. Para la solución se trabaja con una cartografía, la cual contiene al menos una capa de punto. En la capa de punto cada punto será la representación gráfica de un servicio y tendrán toda la información del mismo. Para generar el diagrama de Voronoi se utiliza la capa de punto y se genera una capa regionalizada.

2.4 Requisitos de software

Los requisitos de software son una descripción general de cómo debe funcionar el sistema a implementar. Estos se pueden clasificar en funcionales y no funcionales. Por la metodología escogida los requisitos funcionales se van a representar mediante historias de usuario.

2.4.1 Historias de usuario (HU)

Las Historias de Usuario son la forma en que se especifican en XP las funcionalidades del sistema, realizándose una por cada característica principal del sistema; tienen el mismo propósito que los casos de uso en las metodologías de desarrollo de *software* pesadas, aunque no son lo mismo ya que son escritas por los propios clientes desde su perspectiva del sistema, por lo que serán descripciones cortas y escritas en el lenguaje del usuario. El

Capítulo 2. Descripción de la propuesta de solución

tratamiento de las HU es muy dinámico y flexible, en cualquier momento estas pueden eliminarse, reemplazarse por otras más específicas o generales, añadirse nuevas o ser modificadas. Cada HU es lo suficientemente comprensible y delimitada para que los programadores puedan implementarla en una semana (JEFFRIES *et al.* 2001).

Para definir las historias de usuario es utilizada la siguiente planilla, que contiene todos los datos necesarios para desarrollar la funcionalidad descrita.

HISTORIA DE USUARIO	
Número: número de la HU, incremental en el tiempo	Usuario: el usuario del sistema que utiliza o protagoniza la historia
Nombre de historia: el nombre de la HU, sirve para identificarla fácilmente entre los desarrolladores y los clientes	
Prioridad: qué tan importante es para el cliente	Riesgo en desarrollo: qué tan difícil es para el desarrollador
Estimación técnica: tiempo estimado en semanas que demora el desarrollo de la HU	Iteración asignada: la iteración a la que corresponde
Descripción: la descripción de la historia, detallando las operaciones del usuario y opcionalmente las respuestas del Sistema	

Tabla 1. Ejemplo de la representación de las HU.

La prioridad en el negocio está dada por:

Alta: se le otorga a las HU que resultan fundamentales en el desarrollo y control integral del sistema.

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

Baja: se le otorga a las HU que sirven de ayuda al control de elementos asociados al desarrollo del sistema.

El riesgo de su desarrollo se clasifican en:

Alta: cuando en la implementación de las HU se consideran la posible existencia de errores que lleven a la inoperatividad del código.

Capítulo 2. Descripción de la propuesta de solución

Media: cuando en la implementación de la HU pueden aparecer errores que puedan retrasar la entrega del sistema.

Baja: cuando pueden aparecer errores que serán tratados con relativa facilidad sin que traigan perjuicios para el desarrollo del sistema.

HISTORIA DE USUARIO	
Número: HU-01	Usuario: usuario
Nombre de historia: grafica polígonos de Voronoi tomando como punto generador un servicio dado	
Prioridad: alta	Riesgo en desarrollo: alta
Estimación técnica: 7	Iteración asignada: 1
Descripción: toma la capa donde se encuentran los puntos que hacen referencia a los servicios y se grafica el diagrama de Voronoi correspondiente	

Tabla 2. HU-01.

HISTORIA DE USUARIO	
Número: HU-03	Usuario : usuario
Nombre de historia: ubicar un servicio.	
Prioridad: alta	Riesgo en desarrollo: alta
Estimación técnica: 2	Iteración asignada: 1
Descripción: el usuario dará clic en la ubicación deseada y se le mostrará una ventana donde debe llenar los datos del servicio a ubicar. Luego se indicará la ubicación de dicho servicio en el mapa y se guardarán los datos	

Tabla 3. HU-03.

Capítulo 2. Descripción de la propuesta de solución

2.4.2 Requisitos no funcionales (RNF)

RNF1. Usabilidad

- Las funcionalidades principales están orientadas a íconos para un mayor reconocimiento por parte del usuario.
- El plugin podrá ser usado por personas con conocimientos básicos en el manejo de SIG.

RNF2. Requerimientos de hardware

- El Servidor debe tener como mínimo 4 GB de RAM y 320 GB de disco duro.
- Procesador Core 2 Duo a2.6 o superior.

RNF3. Requerimientos de software

- Sistema operativo GNU/Linux Ubuntu Server 12.04 o superior.
- El servidor debe tener previamente instalado los módulos de Python 2.7.3.

2.5 Tarjetas Clase – Responsabilidad – Colaborador

Para guiar el diseño, la metodología XP usa las tarjetas Cargo o Clase, Responsabilidad y Colaboración (CRC). Esta técnica se usa para guiar el sistema a través del análisis donde las clases se examinan, se filtran y se refinan en base a sus responsabilidades con respecto al sistema. El nombre de la clase se coloca a modo de título en la tarjeta, 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. Las tarjetas determinan el comportamiento de cada actividad.

Clase: Voronoi	
Responsabilidad	Colaboración
- Voronoi	- PyQt4.QtCore - PyQt4.QtGui - qgis.core - VoronoiDialog

Tabla 4. Tarjeta CRC Voronoi.

Capítulo 2. Descripción de la propuesta de solución

Clase: ServicioDialog	
Responsabilidad	Colaboración
<ul style="list-style-type: none">- <code>__init__</code>	<ul style="list-style-type: none">- PyQt4.QtCore- PyQt4.QtGui- NuevoServicioUi

Tabla 5. Tarjeta CRC ServicioDialog.

Clase: NuevoServicio	
Responsabilidad	Colaboración
<ul style="list-style-type: none">- <code>__init__</code>- <code>cargarCombo</code>- <code>validar</code>	<ul style="list-style-type: none">- PyQt4.uic.properties- PyQt4.QtCore- PyQt4.QtGui- qgis.core- Servicio- NuevoServicioUi

Tabla 6. Tarjeta CRC NuevoServicio.

Capítulo 2. Descripción de la propuesta de solución

Clase: <i>VoronoiDialog</i>	
Responsabilidad	Colaboración
- archivoSalida	- PyQt4.QtCore
- run	- PyQt4.QtGui
- cargarCapaPuntos	- qgis.core
- seleccionarServicio	- math
- tipoServicio	- shapely.wkt
- cargarCapaPoligonos	- shapely.wkb
- nombreArchivo	- shapely.ops
- calcularPeso	- osgeo
- puntosExtremos	- processing
- voronoiPonderado	
- divideBisectriz	
- divideRegionArc	
- coordenadasArc	
- distanciaEuclidiana	
- radio	

Tabla 7. Tarjeta CRC *VoronoiDialog*.

Capítulo 2. Descripción de la propuesta de solución

Clase: <i>NuevoServicioUi</i>	
Responsabilidad	Colaboración
<ul style="list-style-type: none">- <code>setupUi</code>- <code>retranslateUi</code>	<ul style="list-style-type: none">- <code>PyQt4.QtCore</code>- <code>PyQt4.QtGui</code>

Tabla 8. Tarjeta CRC *NuevoServicioUi*.

Clase: <i>Servicio</i>	
Responsabilidad	Colaboración
<ul style="list-style-type: none">- <code>insertarServicio</code>- <code>run</code>- <code>crearCapa</code>	<ul style="list-style-type: none">- <code>PyQt4.QtCore</code>- <code>PyQt4.QtGui</code>- <code>qgis.core</code>- <code>qgis.utils</code>- <code>qgis.gui</code>- <code>NuevoServicio</code>- <code>ServicioDialog</code>

Tabla 9. Tarjeta CRC *Servicio*.

Capítulo 2. Descripción de la propuesta de solución

2.6 Diagrama de clases

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación (LARMAN, CRAIG 1999). Los diagramas de este tipo se presentan a continuación en la figura 8 y figura 9.

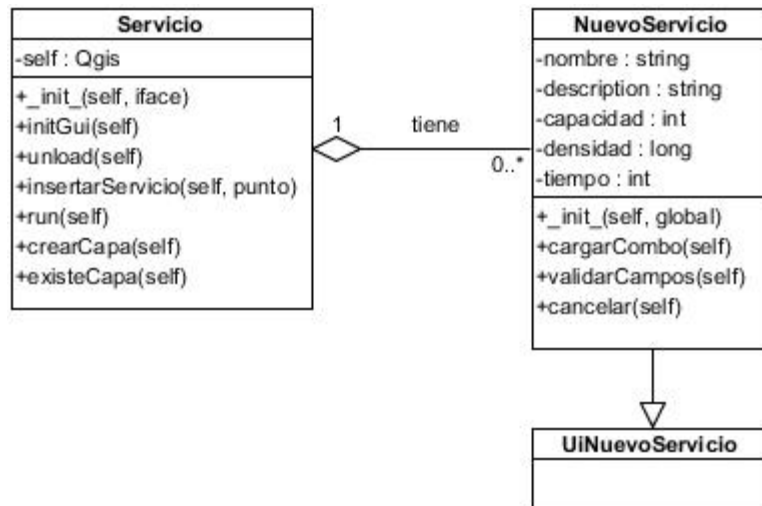


Figura 8. Diagrama de clases de Servicio.

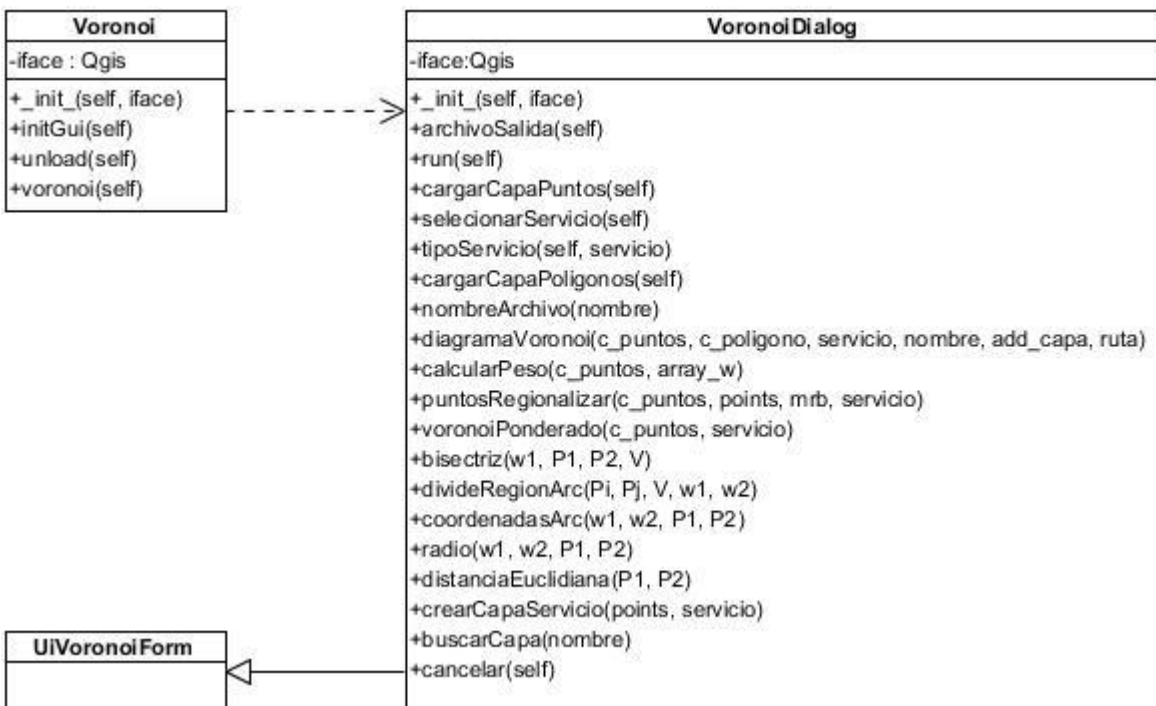


Figura 9. Diagrama de clases de LRSVoronoi.

Capítulo 2. Descripción de la propuesta de solución

2.7 Patrones de diseño

Un patrón es un par problema/solución con nombre que se puede aplicar en nuevos contextos, con consejos acerca de cómo aplicarlo en nuevas situaciones y discusiones sobre sus compromisos. Un patrón es una descripción de un problema bien conocido que suele incluir:

- Descripción.
- Escenario de Uso.
- Solución concreta.
- Las consecuencias de utilizar este patrón.
- Ejemplos de implementación.
- Lista de patrones relacionados(LARMAN, CRAIG 1999; LARMAN, C 2003).

2.7.1 Patrones para asignar responsabilidades (GRASP)

Los patrones GRASP describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en formas de patrones. GRASP es un acrónimo que significa General Responsibility Assignment Software Patterns. El nombre se eligió para indicar la importancia de captar estos principios, si se quiere diseñar eficazmente el software orientado a objetos(LARMAN, CRAIG 1999; LARMAN, C 2003).

Experto en información

La responsabilidad de la creación de un objeto o la implementación de un método, debe recaer sobre la clase que conoce toda la información necesaria para crearlo. A continuación se muestra un ejemplo en la figura 8:



Figura 10. Evidencia de patrón experto.

Capítulo 2. Descripción de la propuesta de solución

Creador

Es el responsable de la creación o instanciación de nuevos objetos o clases. Éste patrón nos dice que la nueva instancia podrá ser creada por una clase si:

- Tiene la información necesaria para realizar la creación del objeto.
- Usa directamente las instancias creadas del objeto.
- Almacena o maneja varias instancias de la clase.
- Contiene o agrega la clase.

A continuación se muestra un ejemplo en la figura 9:

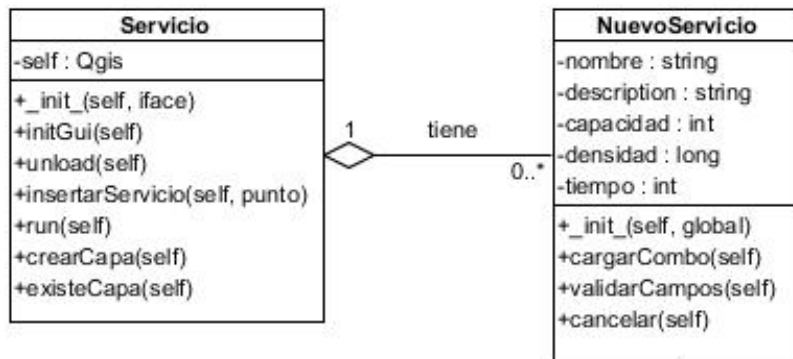


Figura 11. Evidencia del patrón creador.

Bajo Acoplamiento

Es la idea de tener las clases lo menos ligadas entre sí que se pueda. De tal forma que en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, potenciando la reutilización, y disminuyendo la dependencia entre las clases. El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Acoplamiento bajo significa que una clase no depende de muchas clases (LARMAN, CRAIG 1999). En la implementación del plugin es aplicado este patrón en la mayoría de las clases, propiciando que los componentes sean fáciles de entender por separado y de reutilizar.

Alta cohesión

La cohesión es una medida de cuan 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 clase con baja cohesión hace muchas cosas no afines o un trabajo excesivo. No conviene este tipo de clases pues son difíciles de comprender, reutilizar y conservar.

Capítulo 2. Descripción de la propuesta de solución

Conclusiones del capítulo

El estudio de los diagramas de Voronoi ponderados realizado en el capítulo 1 permitió el desarrollo del algoritmo LRSVoronoi para darle solución al problema identificado. Además, los artefactos definidos por la metodología XP y los diagramas de clases correspondientes, serán las guías para proceso de implementación de la solución.

Capítulo 3. Implementación y prueba

CAPÍTULO 3. IMPLEMENTACIÓN Y PRUEBA

En este capítulo se abordaron dos de las fases más importantes en el ciclo de vida de cualquier software: Implementación y Pruebas, en esta fase se materializa el software mediante su codificación y se realizaron las pruebas necesarias para comprobar si fueron cumplidos los objetivos trazados.

3.1 Implementación

Estándar de programación

XP hace énfasis en que la comunicación de los programadores es a través del código, con lo cual es indispensable que se sigan ciertos estándares de programación para mantener el código legible (GRUPO 2003). Esto permite que cualquier integrante del equipo de desarrollo pueda modificar cualquier parte del código, además que sea un código entendible para otros programadores.

El estándar que se utiliza en el desarrollo de la solución es:

Máxima longitud de las líneas

- Todas las líneas se limitan a un máximo de 80 caracteres.

Importaciones

- Las importaciones se encuentran en líneas separadas.

Comentarios

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

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

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

Estilo de los nombres

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

Ejemplo: NuevoServicio

Capítulo 3. Implementación y prueba

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

Ejemplo: tipoServicio

3.2 Pruebas de software

Las pruebas de software tienen como objetivos detectar errores no encontrados hasta el momento en la aplicación. Se puede hablar entonces del éxito de las pruebas siempre y cuando se hallen errores. Con las pruebas se puede además observar hasta qué punto el software parece funcionar en concordancia con las HU descritas. Comprobar el funcionamiento de programas puede convincentemente demostrar la presencia de errores, pero nunca puede demostrar su ausencia (PRESSMAN 1997).

Para comprobar el correcto funcionamiento del sistema serán utilizados los métodos de prueba de caja blanca y las de aceptación o pruebas funcionales.

3.2.1 Pruebas de caja blanca

La prueba de la caja blanca es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivar los casos de prueba.

Las pruebas de caja blanca intentan garantizar que:

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

El método del camino básico (MCCABE 1976) permite obtener una medida de la complejidad de un diseño procedimental, y utilizar esta medida como guía para la definición de una serie de caminos básicos de ejecución, diseñando casos de prueba que garanticen que cada camino se ejecuta al menos una vez. Este método se representa a través de un grafo de flujo donde:

Capítulo 3. Implementación y prueba

- Cada nodo representa una o más sentencias procedimentales.
- Un solo nodo puede corresponder a una secuencia de pasos del proceso y a una decisión.
- Las flechas (aristas) representan el flujo de control.

Complejidad ciclomática

Es una medida que proporciona una idea de la complejidad lógica de un programa.

- La complejidad ciclomática coincide con el número de regiones del grafo de flujo.
- La complejidad ciclomática $V(G)$, de un grafo de flujo G , se define como:

$$V(G) = \text{Aristas} - \text{Nodos} + 2$$

- La complejidad ciclomática $V(G)$, de un grafo de flujo G , también se define como:

$$V(G) = \text{Nodos de predicado} + 1$$

A partir del valor de la complejidad ciclomática obtenemos el número de caminos independientes, que nos dan un valor límite para el número de pruebas que tenemos que diseñar.

La complejidad ciclomática se le calculó al método **VoronoiPonderado()** de la solución, el cual se muestra en la figura 12.

A partir del grafo de flujo de la figura 13, la complejidad ciclomática sería:

- Como el grafo tiene 4 regiones, $V(G) = 4$
- Como el grafo tiene 13 aristas y 11 nodos, $V(G) = 13 - 11 + 2 = 4$
- Como el grafo tiene 3 nodos predicado, $V(G) = 3 + 1 = 4$

Arribando a 4 caminos linealmente independientes, los mismos se muestran a continuación:

1. 1-2-11
2. 1-2-3-4-10-2-11
3. 1-2-3-4-5-6-7-9-4-10-2-11
4. 1-2-3-4-5-6-8-9-4-10-2-11

Capítulo 3. Implementación y prueba

```
def voronoiPonderado(layer,servicio):
    points = []
    w=[]
    mrb=[0,0,0,0]
    puntosExtremos(layer,points,mrb,servicio)
    calcularPeso(points,w)
    V=[]
    Vtemp=[]
    offset = (mrb[1] - mrb[0]) * 0.3
    Vtemp.append(QgsPoint(mrb[0]-offset,mrb[2]-offset))
    Vtemp.append(QgsPoint(mrb[0]-offset,mrb[3]+offset))
    Vtemp.append(QgsPoint(mrb[1]+offset,mrb[3]+offset))
    Vtemp.append(QgsPoint(mrb[1]+offset,mrb[2]-offset))
    Vtemp.append(QgsPoint(mrb[0]-offset,mrb[2]-offset))
    V.append(QgsGeometry.fromPolygon([Vtemp]))

    for j in range(len(points)-1):
        Pj=points[j]
        polig=V.pop(j)
        wj=w[j]
        XX=[]
        for i in range(j+1,len(points)):
            Pi=points[i]
            wi=w[i]
            geo=QgsGeometry()
            if wi==wj:
                geo= bisectriz(wj,Pj,Pi,polig)
            else:
                geo= divideRegionArc(Pi,Pj,polig,wi,wj)
            XX.append(geo)
        geo1=QgsGeometry.unaryUnion(XX)
        V.insert(j,polig.difference(geo1))
        V.append(geo1)

    return V
```

The diagram shows line numbers 1 through 11 on the right side of the code. Arrows point from these numbers to specific lines in the code. Brackets on the right side group lines into larger sections:

- Lines 1 through 11 are grouped together by a large bracket on the right.
- Lines 1 through 5 are grouped together by a bracket on the right.
- Lines 2 through 3 are grouped together by a bracket on the right.
- Lines 4 through 5 are grouped together by a bracket on the right.
- Lines 6 through 8 are grouped together by a bracket on the right.
- Lines 9 through 10 are grouped together by a bracket on the right.

Figura 12. Código del método *VoronoiPonderado()*.

Capítulo 3. Implementación y prueba

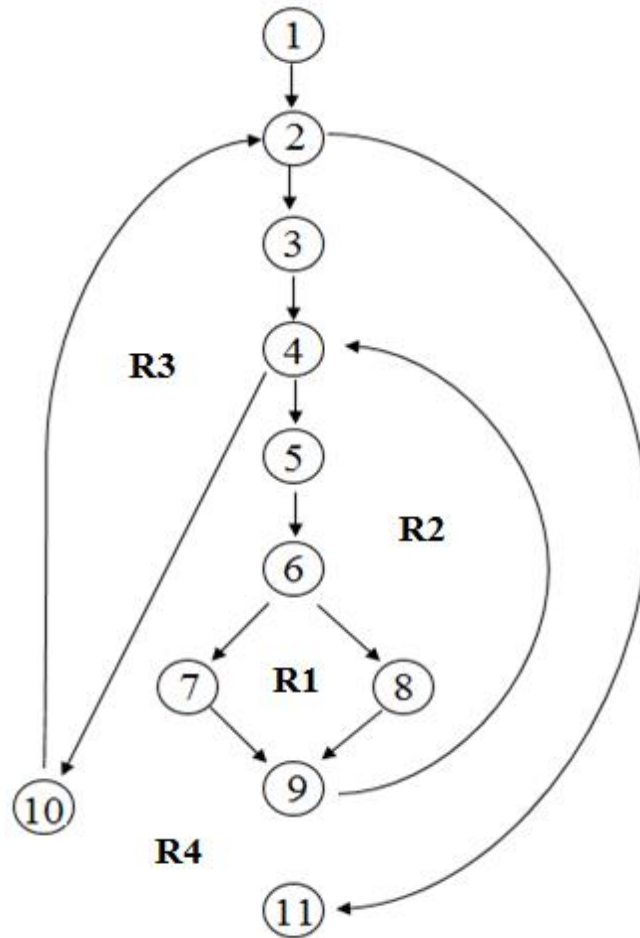


Figura 13. Grafo de flujo.

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

Caso de prueba para el camino básico 1 (1-2-11)	
Descripción	Al no tener puntos para analizar, devuelve el arreglo de polígonos vacío
Condición de ejecución	Arreglo de puntos vacío
Entrada	Capa de puntos y el tipo de servicio
Resultado	Retorna un arreglo vacío
Resultado de la prueba	Satisfactorio

Tabla 10. Caso de prueba camino básico 1.

Capítulo 3. Implementación y prueba

Caso de prueba para el camino básico 2 (1-2-3-4-10-2-11)	
Descripción	Analiza los puntos y al tener un solo punto no tiene elementos con quien comparar
Condición de ejecución	Arreglo de puntos con un solo elemento
Entrada	Capa de puntos y el tipo de servicio
Resultado	Retorna un arreglo con un solo polígono
Resultado de la prueba	Satisfactorio

Tabla 11. Caso de prueba camino básico 2.

Caso de prueba para el camino básico 3 (1-2-3-4-5-6-7-9-4-10-2-11)	
Descripción	Analiza los puntos y traza bisectrices entre puntos para generar los polígonos
Condición de ejecución	Los puntos poseen pesos iguales
Entrada	Capa de puntos y el tipo de servicio
Resultado	Retorna un arreglo de polígonos
Resultado de la prueba	Satisfactorio

Tabla 12. Caso de prueba camino básico 3.

Caso de prueba para el camino básico 4 (1-2-3-4-5-6-8-9-4-10-2-11)	
Descripción	Analiza los puntos y realiza arcos en dependencia de los pesos de los mismo para generar los polígonos
Condición de ejecución	Los puntos poseen pesos diferentes
Entrada	Capa de puntos y el tipo de servicio
Resultado	Retorna un arreglo de polígonos
Resultado de la prueba	Satisfactorio

Tabla 13. Caso de prueba camino básico 4.

Capítulo 3. Implementación y prueba

3.2.2 Pruebas de aceptación

Dentro de las pruebas de aceptación se utilizan las pruebas de caja negra que se crean a partir de las HU y son realizadas por el cliente y los usuarios finales de la aplicación. En ellas serán probadas las funcionalidades exigidas por el cliente. Luego de haber superado las pruebas de aceptación podrá considerarse que la aplicación es apta para el uso y despliegue.

Para eliminar la influencia de conflictos de intereses, y para que sea lo más objetiva posible, la prueba de aceptación nunca debería ser responsabilidad de los ingenieros de software que han desarrollado el producto. Para la preparación, la ejecución y la evaluación de la prueba de aceptación ni siquiera hacen falta conocimientos informáticos. Sin embargo, un conocimiento amplio de métodos y técnicas de prueba y de la gestión de la calidad en general facilita esta labor (PRUEBAS DE SOFTWARE).

Las pruebas de aceptación se realizaron de la forma siguiente:

1. Se redactan los casos de prueba teniendo en cuenta el orden de las HU y los niveles de prioridad.
2. Se hace la planificación con el cliente de cuándo y cuáles pruebas son llevadas a cabo.
3. Se reúnen los miembros del proyecto seleccionados para realizar las pruebas.
4. Se completan cada uno de los campos de la tabla de pruebas de aceptación con el resultado.

La persona adecuada (o el equipo adecuado) para llevar a cabo la prueba de aceptación dispone de estos conocimientos y además es capaz de interpretar los requerimientos especificados por los futuros usuarios del sistema del software en cuestión.

Capítulo 3. Implementación y prueba

Caso de prueba de aceptación	
Código: HU-01-P	Historia de usuario: HU-01
Nombre: graficar polígonos de Voronoi tomando como punto generador un servicio dado.	
Descripción: el usuario escogerá la capa de los puntos y el tipo de servicio al cual desea realizarle el diagrama de Voronoi, además del nombre y la ubicación de donde desea guardar la capa	
Condiciones de ejecución: debe existir una capa de puntos disponible, que contenga los datos del servicio, dígame capacidad de atención, tiempo promedio del servicio y la densidad poblacional del lugar donde está ubicado	
Resultados esperados: se genera y muestra un mapa con la representación de los servicios y el diagrama de Voronoi correspondiente a ellos	
Evaluación de la prueba: Satisfactoria	

Tabla 14. Caso de prueba aceptación HU-01.

Caso de prueba de aceptación	
Código: HU-02-P	Historia de usuario: HU-02
Nombre: ubicar un servicio	
Descripción: el usuario selecciona la funcionalidad y al dar clic en la parte del mapa donde se desee agregar el servicio se le muestra un formulario para que introduzca los datos	
Condiciones de ejecución: no debe existir otro servicio con el mismo nombre	
Resultados esperados: se inserta un nuevo servicio a la capa de punto	
Evaluación de la prueba: satisfactoria	

Tabla 15. Caso de prueba aceptación HU-02.

Capítulo 3. Implementación y prueba

3.3 Caso de estudio

Para constatar los resultados de la solución propuesta, se aplicó un caso de estudio tomando como muestra a la provincia de La Habana, seleccionando de ella un conjunto de hospitales, policlínicos y clínicas estomatológicas que se muestran en las tablas 16, 17 y 18 respectivamente, cada uno con sus indicadores. Los indicadores a tener en cuenta para el cálculo del peso de cada servicio va estar conformado por: la población de la región (Dp) donde se encuentra ubicado el servicio, la capacidad del servicio (Cp), entiéndase esto como la cantidad de persona para las cuales está diseñado el servicio y por último el tiempo del servicio o tiempo promedio que demora en ser atendido un paciente (Tp), el cual se obtuvo mediante la técnica de observación. Partiendo de la ecuación de la velocidad $v = \frac{S}{t}$ se define como peso (P), a la rapidez con que puede ser atendida toda la población de la región donde se encuentra el servicio. De ahí que:

$$T = Dp * Tp$$

y

$$S = Cp$$

Despejando en la ecuación de velocidad se obtiene:

$$P = \frac{Cp}{Dp * Tp}$$

Nombre de clínicas	Capacidad de servicio	Población	Tiempo promedio
Cira García	70	1500	10
CIPBIN	60	1230	17
CIREN	68	1000	13
Salvador Allende	90	1200	18
Ramón López	75	1100	15
13 y 3ra	70	1030	10
Elpidio Berovides	65	1070	14
H y 21	65	980	13
Los Pinos	78	1200	12
Santa Catalina	70	1200	15

Tabla 16. Servicios de clínicas estomatológicas.

Capítulo 3. Implementación y prueba

Nombre de hospital	Capacidad de servicio	Población	Tiempo promedio
Hermanos Almeijeiras	600	2500	15
Calixto García	450	1800	20
Manuel Fajardo	500	1800	12
Ramón González	550	2000	10
Salvador Allende	600	2100	12
Joaquín Albarrán	400	1400	14
Pando Ferrer	520	1700	14
Frank País	700	2800	15
Wiliam Soler	700	3000	13
Satos Suárez	480	2300	16

Tabla 17. Servicios de hospitales.

Nombre de policlínico	Capacidad de servicio	Población	Tiempo promedio
Carlos J. Finlay	145	1040	13
Cristóbal Labra	124	1150	17
5 de Septiembre	114	1250	18
15 y 18	110	1120	12
19 de Abril	122	1240	13
26 de Julio	134	1230	18
28 de Enero	135	1060	16
30 de Noviembre	115	1300	15
Máximo Gómez	120	1180	12

Tabla 18. Servicios de policlínicos.

Capítulo 3. Implementación y prueba

A continuación se muestra los resultados de aplicar la solución al caso de estudio.

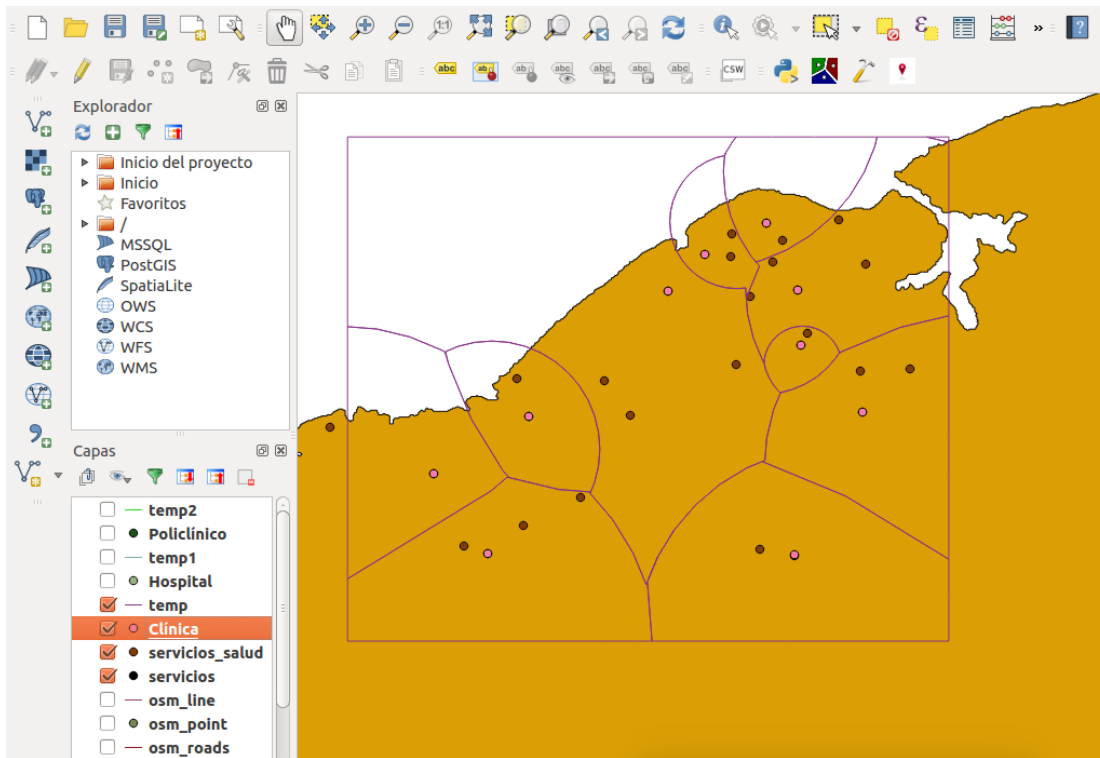


Figura 14. Regionalización del servicio de clínicas estomatológicas.

Capítulo 3. Implementación y prueba

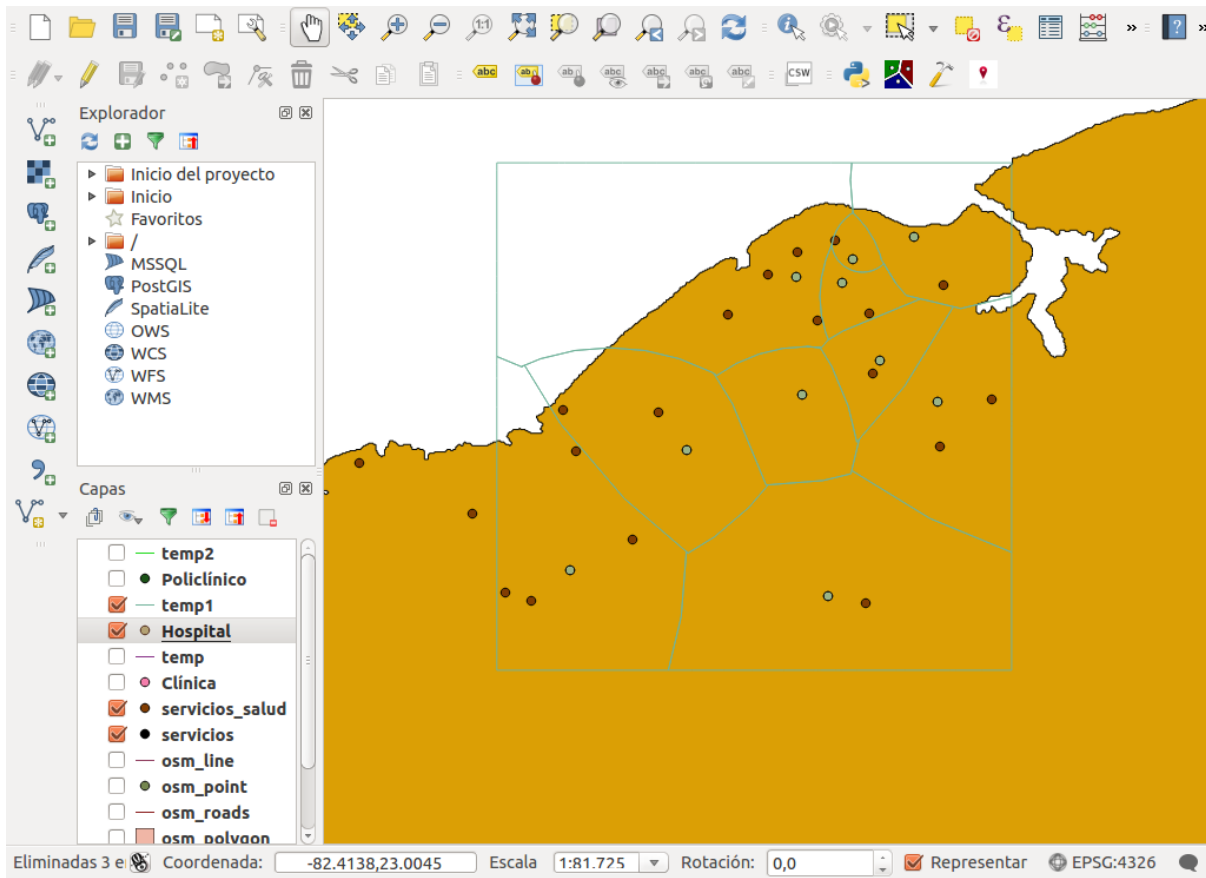


Figura 15. Regionalización del servicio de hospitales.

Capítulo 3. Implementación y prueba

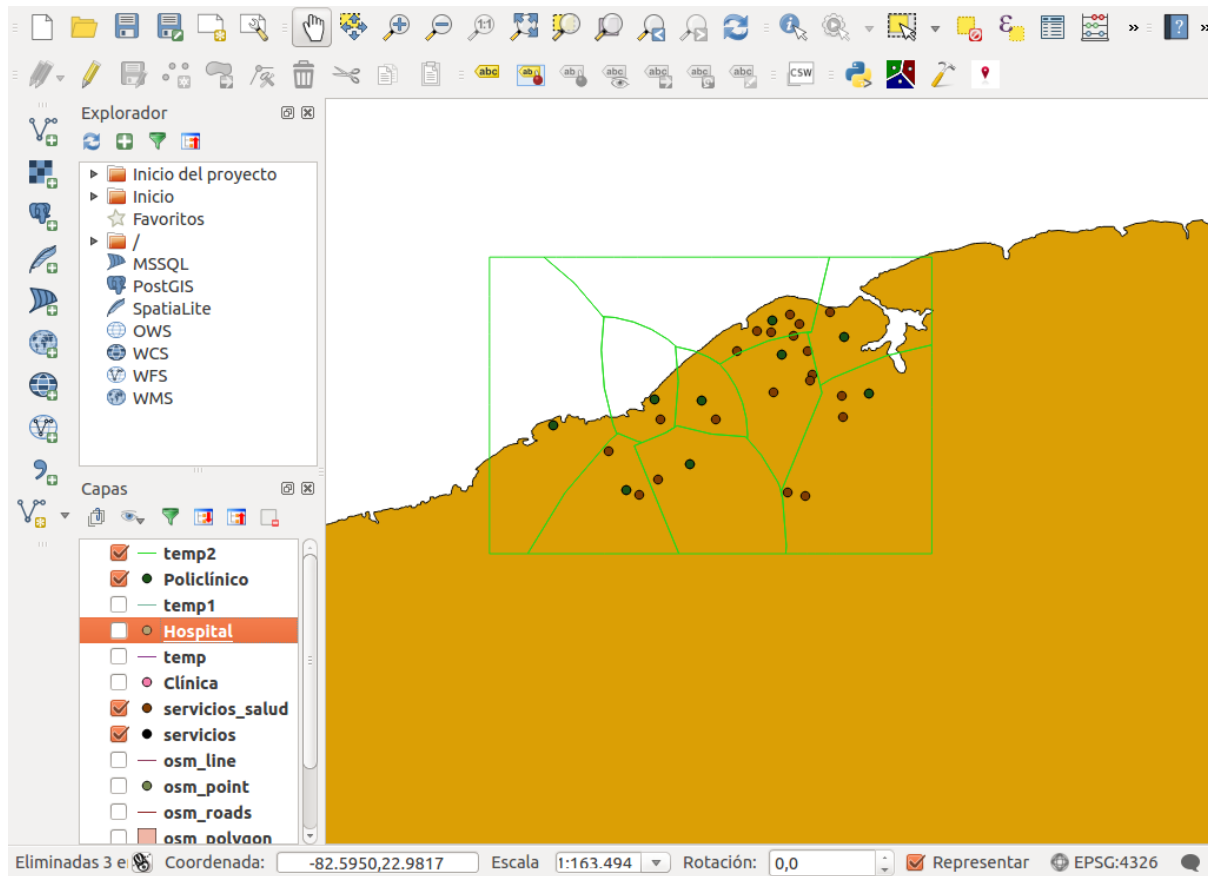


Figura 16. Regionalización del servicio de policlínicos.

Conclusiones del capítulo

Luego de implementada la solución y realizadas las pruebas de caja blanca y aceptación, se comprobó que existe una correspondencia positiva en los resultados obtenidos de las pruebas. La aplicación de la solución a un caso de estudio, mostró resultados satisfactorios, cumpliendo con las expectativas y los requisitos previstos.

Conclusiones generales

Conclusiones generales

Con la realización del presente trabajo de diploma se logró cumplir con los objetivos planteados, obteniéndose una solución informática para la regionalización y localización de servicios. A partir de los resultados obtenidos se puede concluir que:

- Las alternativas existente para la solución al problema de la localización de servicios tienen limitaciones cuando se desea saber dónde debe recibir un determinado servicio una región pues están a desde un punto determinar cuál es el servicio óptimo. Los diagramas de Voronoi son una alternativa para resolver este problema por ser un método eficiente para la regionalización sin embargo no integran indicadores que pueden incidir sobre el servicio
- La utilización de los diagramas de Voronoi con un peso asociado permitió regionalizar servicios teniendo en cuenta indicadores de variada naturaleza lo que facilita la localización del servicio que debe recibir una región determinada.
- Los resultados obtenidos después de aplicadas las pruebas y la aplicación a un caso de estudio, evidencian que la solución propuesta cumple con los requisitos previstos y que puede ser utilizado como sistema de ayuda a la toma de decisiones en el proceso de regionalización y localización de servicios de salud.

Recomendaciones

Recomendaciones

Una vez vencidos los objetivos de la investigación y teniendo en cuenta las experiencias obtenidas a lo largo de su desarrollo, se recomienda:

- Realizar un análisis para el cálculo de los pesos para cada servicio, en caso de que se quieran incorporar más indicadores.
- Darle continuidad a la investigación mediante la incorporación de nuevos métodos teniendo en cuenta otros criterios de análisis como pueden ser las redes viales y los medios de transporte.

BIBLIOGRAFÍA

- ALFONZO, J. E. M. *Los servicios y la economía mundial - Opinión - EL UNIVERSAL*, 2011. [Disponible en: http://www.eluniversal.com/2011/02/03/opi_art_los-servicios--y-la_03A5094931]
- ALONSO, D. *Novedades de QGIS 2.8 Wien*, 2015. [Disponible en: <http://mappinggis.com/2015/02/novedades-de-qgis-2-8-wien/>]
- ANDERSON, S. *An evaluation of spatial interpolation methods on air temperature in Phoenix, AZ*, 2002.
- ARRIAGADA, I. *Abriendo la caja negra del sector servicios en Chile y Uruguay Género, familias y trabajo: rupturas y continuidades. Desafíos para la investigación política. Buenos Aires. CLACSO, Consejo Latinoamericano de Ciencias Sociales. Agosto, 2007.*
- ASH, P. F. and E. D. BOLKER Generalized dirichlet tessellations *Geometriae Dedicata*, 1986, 20(2): 209-243.
- BECK, K. *Extreme programming explained: embrace change*. Addison-Wesley Professional, 2000. p. 0201616416
- BOOCH, G.; J. RUMBAUGH, et al. *El lenguaje unificado de modelado*. Addison-Wesley, 1999. p.
- BRAVO, R. B. *Localización de servicios en el plano mediante técnicas de optimización dc*. Universidad de Sevilla, 1999. p. 8468977454
- BRUNO, N. R. *Diagrama de dominio*, 2014. [Disponible en: http://es.slideshare.net/cristhianke/diagrama-de-dominio-armando?gid=fcec4cd6-75ad-4f72-b572-94414ab5788b&v=qf1&b=&from_search=27]
- CAMPOS, A. P. *El sistema de información geográfica: un instrumento para la planificación y gestión urbana Geographicalia*, 1991, (28): 175-192.
- CARDOZO, O. D.; E. L. GÓMEZ, et al. *Teoría de grafos y sistemas de información geográfica aplicados al transporte público de pasajeros en resistencia (Argentina) Revista Transporte y Territorio*, 2009, (1): 89-111.
- CHASCO YRIGOYEN, M. D. C. *Econometría espacial aplicada a la predicción-extrapolación de datos microterritoriales*, 2002.
- CHICA OLMO, M.; J. D. GARCÍA, et al. *Introducción al Análisis Geoestadístico de variables espaciales Logroño. 308p*, 1995.
- DE BERG, M.; M. VAN KREVELD, et al. *Computational geometry*. Springer, 2000. p. 3662042479
- DEZA, M.-M. and E. DEZA. *Dictionary of distances*. Elsevier, 2006. p. 0080465544
- DUQUE, R. G. *Python para todos*, 2011.
- ESCOBAR MARTÍNEZ, F. *Los Sistemas de Información Geográfica en la localización de servicios. Centros de Salud y Clubes de Jubilados de Alcalá de Henares*, Tesis Doctoral. Universidad de Alcalá. Alcalá de Henares, España, 1995. p.

Bibliografía

- FERNÁNDEZ NÚÑEZ, H. M. SIG-ESAC: Sistema de Información Geográfica para la gestión de la estadística de salud de Cuba *Revista Cubana de Higiene y Epidemiología*, 2006, 44(3): 0-0.
- GRUPO, I. *Metodologías ágiles en el desarrollo de software*, Alicante: Patricio Letelier Torres, Emilio A. Sánchez, 2003.
- I SUBIRANA, J. C. Infraestructura de Datos Espaciales (IDE). Definición y desarrollo actual en España *Scripta Nova: revista electrónica de geografía y ciencias sociales*, 2004, (8): 61.
- ILLERIS, S. Localización de los servicios a empresas en zonas urbanas y regionales *Economía Industrial*, 1997, (313): 93-103.
- JEFFRIES, R.; A. ANDERSON, et al. *Extreme programming installed*. Addison-Wesley Professional, 2001. p. 0201708426
- JETBRAINS. *Python IDE & Django IDE for Web developers : JetBrains PyCharm*, 2014. [Disponible en: www.jetbrains.com/pycharm]
- KAWAMURA, H.; T. SASAKI, et al. *Spline interpolation method*, Google Patents, 1992.
- KERRIGAN, M.; A. MOCAN, et al. The web service modeling toolkit-an integrated development environment for semantic web services. en: *The Semantic Web: Research and Applications*. Springer, 2007.789-798.p.
- LARMAN, C. *UML y Patronos*. Pearson, 1999. p. 8420534382
- LARMAN, C. *UML Y PATRONES. Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Aragón DF, Madrid: Pearson Educación. SA, 2003.
- LETELIER, P. *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*, 2006.
- LIMITED, R. C. *Using Qt Designer*, 2014. [Disponible en: <http://pyqt.sourceforge.net/Docs/PyQt4/designer.html>]
- LONGLEY, P. A.; M. F. GOODCHILD, et al. *Geographic information system and Science England: John Wiley & Sons, Ltd*, 2001.
- LOUDEN, K. C. *Lenguajes de programación: Principios y práctica*. Cengage Learning Latin America, 2004. p. 9706862846
- MATHEWS, J. H.; K. D. FINK, et al. *Métodos numéricos con MATLAB*. Prentice Hall, 2000. p.
- MCCABE, T. J. A complexity measure *Software Engineering, IEEE Transactions on*, 1976, (4): 308-320.
- MIGNONE, P. Arqueología y SIG histórico: desafíos interpretativos del «Itinerario» del Oidor de la Real Audiencia de Charcas Juan de Matienzo (1566) a la luz de la arqueología espacial *Arqueología iberoamericana*, 2013, 17: 23-32.
- MIRANDA-SALAS, M. and A. R. CONDAL Importancia del análisis estadístico exploratorio en el proceso de interpolación espacial: caso de estudio Reserva Forestal Valdivia *Bosque (Valdivia)*, 2003, 24(2): 29-42.

Bibliografía

- MURDICK, R. G.; B. RENDER, *et al.* *Service operations management*. Allyn and Bacon Boston, MA, 1990. p.
- MURILLO, D. M.; I. ORTEGA, *et al.* Comparación de métodos de interpolación para la generación de mapas de ruido en entornos urbanos *Revista Ingenierías USBMed*, 2012, 3(1): 62-68.
- OKABE, A.; B. BOOTS, *et al.* *Spatial tessellations: concepts and algorithms of Voronoi diagrams. J*, Wiley & Sons, Chichester [etc.], 2000.
- . *Spatial tessellations: concepts and applications of Voronoi diagrams*. John Wiley & Sons, 2009. p. 047031785X
- PARADIGM, V. Visual paradigm for uml *Visual Paradigm for UML-UML tool for software application development*, 2010.
- PARDOS, E.; F. R. MOROLLÓN, *et al.* *La localización de los servicios empresariales en Aragón: determinantes y efectos*. Fundación Economía Aragonesa, 2004. p.
- PESQUER, L.; J. MASÓ, *et al.* Integración SIG de regresión multivariante, interpolación de residuos y validación para la generación de rásters continuos de variables meteorológicas *Revista de Teledetección*, 2007, 28: 69-76.
- PRESSMAN, R. S. *Ingeniería del Software: Un enfoque práctico*. Mikel Angoar, 1997. p. 8448111869
- PRUEBASDESFTWARE. *La prueba de aceptación es la prueba más importante para los productos software*. Disponible en: <http://pruebasdesoftware.com/pruebadeaceptacion.htm>
- PUEBLA, J. G. and M. GOULD. *SIG: Sistemas de información geográfica*. Síntesis, 1994. p. 8477382468
- RAISZ, E. and J. M. MANTERO. *Cartografía general*. Omega, 1985. p. 8428200076
- RODRÍGUEZ GAVILÁN, G. Representación gráfica de superficies mediante mapas térmicos, 2009.
- SEGUÍ PONS, J.; M. RUIZ PÉREZ, *et al.* LA PLANIFICACIÓN DE RUTAS DE TRANSPORTE ESCOLAR A TRAVÉS DE UN SIG: EL PROYECTO SIGTEBAL, 2003.
- SONG, T.; C. XIMIN, *et al.* Vector-based Realization of Multiplicatively Weighted Voronoi Diagrams with ArcGIS Engine, 2014.
- SUGIHARA, K.; A. OKABE, *et al.* *Spatial tessellations: concepts and applications of Voronoi diagrams Probability and Statistics*, 1992.
- SUMMERFIELD, M. *Rapid GUI programming with Python and Qt: the definitive guide to PyQt programming*. Pearson Education, 2007. p. 0132703068
- TEAM, Q. G. D. Quantum GIS geographic information system, 2012.
- VORONOI, G. Nouvelles applications des paramètres continus à la théorie des formes quadratiques. Premier mémoire. Sur quelques propriétés des formes quadratiques positives parfaites *Journal für die reine und angewandte Mathematik*, 1908, 133: 97-178.