

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Facultad 6



TÍTULO: Sistema de Información Geográfica para el análisis espacial de la distribución de los productos gastronómicos de la Universidad de las Ciencias Informáticas

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

AUTOR: Lisandra Lucia Sosa Nuñez

TUTOR: Ing. Yoandry Lazo Nodarse

Ciudad de La Habana

2011

*"Sueña y serás libre de espíritu, lucha y serás
libre en la vida."*

le



Dedico este trabajo a las personas más importantes de mi existencia, las cuales me ayudan a no perder mi eje, porque sólo pensarlos me da fuerzas para enfrentarme a cualquier reto que me depare el destino. Porque me aman, me cuidan y me guían, por eso y por todas las cosas que me han dado, a mis queridos padres Leticia y Alfonso muchas gracias por todo, los amo.

Otra de las personas más importantes de mi vida, la cual le debo todo lo que soy es a mi abuelo Nuñez. A ti abuelo te dedico esta investigación, porque tú me enseñaste a contar, a leer y a escribir. Sé que hoy soy tu mayor orgullo, porque entre todos tus nietos soy la primera universitaria. Te quiero mucho.

Antes que todo a las personas que han sido mi faro guía durante todos estos largos años, sin ellas no sería posible ser lo que soy, mis más grandes agradecimientos, para esos seres que sacrifican todo sin esperar nada a cambio, mis padres. Gracias mami y papi por depositar toda su confianza en que yo iba a ser alguien en la vida, por el apoyo incondicional y el infinito amor, porque este es su sueño hecho realidad pues lo que soy se lo debo a ustedes, gracias más que nada por regalarme la vida.

A mis abuelos Mercedes y Nunez, gracias por siempre estar ahí para mí, por la educación que me dieron durante toda mi vida y por quererme tanto. No los olvidaré jamás.

A mis abuelos Carlota y Alfonso, por permitirme hacer todas las cosas que siempre quise. Gracias.

A mis dos adorados tíos, mejores que ellos no los quiero. A mi tía Lourdes por darme la oportunidad de tener más de una madre y por ser ejemplo importante en mi vida. A mi tío Pepe por apoyar todas mis locuras y guardar todos mis secretos.

A mis queridos primos que de una forma u otra forman parte de mi vida: Frank, David, Carlos y Pepitín. Gracias por estar ahí cuando los necesito.

A mis padrinos Ileana y Toni, por aguantar mis malcriadeces y apoyarme en todo momento, a Marlen por su cariño y por quererme como a una nieta. A Claudia y a Yaniel por ser como unos hermanos para mí y por brindarme su amor. A Maicel por ayudarme en el peor momento de mi vida. A todos ustedes que son mi segunda familia muchas gracias y los quiero mucho.

A mis vecinos locos Ania y Pepe, gracias por complacerme tanto.

Agradecer a una persona en especial, que ha sabido ser amiga y hermana en todo momento de estos años de universidad, que ha estado en mis malos y buenos momentos, que sé que cualquier día va a estar ahí para tenderme una mano, a ti mi pelusita (Anarvis) gracias por tu cariño y comprensión. Sabes que te quiero mucho.

A mi gran amigo Frank Emilio por ser mi hermanito y por no permitir que dejara la escuela. Te debo mucho, te quiero.

A una persona que sin ella no hubiera sido posible realizar esta investigación, a ti Adrián por ayudarme a tirar tanto código, por ser como eres conmigo y por pegarme tu locura, gracias lokjus. Te quiero mucho.

A mi tutor por todo el apoyo que me ha brindado y por leerse tantas veces mi documento. Le deseo mucha suerte en su vida y que logre cumplir todos sus sueños.

Agradezco al tribunal y al oponente de tesis por sus precisas acotaciones que le dieron a este trabajo mayor calidad y rigor científico.

A Jorge, Alberto y Romanuel por ayudarme en esta investigación y por estar siempre dispuestos cuando los necesité. Gracias por todo.

A Yasnary y Jose Ángel por brindarme su amistad y su cariño. Los quiero.

A mis mejores amigas que siempre han estado presentes cuando las he necesitado.

A Leydis, Dayana, Yanelis, Danieyis y Gelsys. Gracias por todo las quiero mucho.

A las killers Misbel y Sallet, por estar ahí cuando uno menos se lo espera, en tan poco tiempo supieron ganarse un lugar en mi corazón, las quiero mucho.

A Sergio por estar conmigo en todo momento y apoyarme en mis días difíciles. Gracias por existir.

A Sandy y a Yurisnel por apoyarme durante estos largos años de mi carrera y por tener siempre una respuesta a mis preguntas. Los recordaré siempre.

A todos esos buenos amigos que he conocido en estos cinco años, mencionarlos sería interminable. A mis compañeros de aula durante todo este tiempo. Gracias.

A mis profes de toda la carrera, por haberme enseñado a aprender. Muchos podrán ser los conocimientos olvidados después de los años pero la metodología del aprendizaje siempre quedará, y eso se lo debo a Uds.

A todas aquellas personas que alguna vez me preguntaron cómo me iba en la tesis y a los que me han ayudado de una forma u otra a realizar este sueño... Gracias!!!

Declaración de Autoría

Declaración de Autoría

Ciudad de La Habana, Julio, 2011.

“Año x de la Revolución”

Yo: Lisandra Lucia Sosa Nuñez declaro que soy la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del 2010.

Lisandra Lucia Sosa Nuñez.

Ing. Yoandry Lazo Nodarse

FIRMA DEL AUTOR

FIRMA DEL TUTOR

Datos de Contacto



Autor

Nombre: Lisandra Lucía

Apellidos: Sosa Nuñez.

Correo electrónico: llsosa@estudiantes.uci.cu

Tutor

Nombre: Yoandry

Apellidos: Lazo Nodarse.

Correo electrónico: ylazo@uci.cu

Una correcta distribución de los alimentos hacia las distintas unidades de servicio de la Universidad de las Ciencias Informáticas es de vital importancia para todos sus habitantes. Los productos gastronómicos se distribuyen actualmente de manera ineficiente, las cafeterías reciben productos sin tener en cuenta la demanda de las mismas y existe un gasto innecesario de combustible en la distribución de estos productos, debido a que no se valoran los elementos referentes a la distancia de las cafeterías.

Como solución a esta problemática se desarrolló un Sistema de Información Geográfica que permite realizar el análisis espacial de la distribución de los productos gastronómicos de la Universidad de las Ciencias Informáticas, facilitando así una aplicación para apoyar el mejoramiento en el área de los servicios gastronómicos de la universidad. Durante el progreso de la investigación se estudiaron algunos conceptos que fueron de gran ayuda para un mejor entendimiento de la solución. Se analizaron las herramientas que fueron utilizadas para el desarrollo del sistema y finalmente se realizó la documentación del ciclo de vida del proyecto.

Palabras Claves: Sistema de Información Geográfica, distribución, análisis espacial.

Proper distribution of food to a variety of service units in the University of Informatics Sciences (UCI) is vital to all its inhabitants. Gastronomic products are currently distributed but inefficiently. Cafeterias receive some of these products without taking into account their demand. There is also a waste of fuel during their distribution because of some elements; the distance among the cafeterias is not taken into consideration.

To solve these problems a Geographic Information System was developed. It allows the real space analysis of the gastronomic products distribution in the university. This application facilitates the upgrading of the gastronomic services in the campus. During the research process some concepts were studied that helped authors to a better understanding of the problem and its solution. Some informatics tools were also analyzed for the development of this application. The documentation of the project life cycle was finally made.

Keywords: Geographic Information System, distribution, space analysis.

Introducción	1
Capítulo 1	6
Fundamentación Teórica.	
1.1 Introducción.....	6
1.2 Conceptos asociados al dominio del problema.....	6
1.2.1. Sistemas de Información Geográfica.	6
1.2.2. Redes Lineales.....	7
1.2.2.1. Red de Flujo.....	7
1.2.2.1.1. Modelos de Redes.	8
1.2.3. Distribución.....	11
1.2.4. Mapas Temáticos.	11
1.2.4.1. Tipos de Mapas Temáticos.....	12
1.3. Análisis de otras soluciones existentes.....	14
1.3.1. Quantum GIS.	15
1.3.2. ArcGIS.....	15
1.3.3. ArcView.	16
1.3.4. ArcEditor.....	16
1.3.5. ArcInfo.....	17
1.3.6. GRASS.	17
1.4. Conclusiones Parciales.....	18
Capítulo 2	20
Tecnologías y herramientas a utilizar.	
2.1. Introducción.....	20
2.2. Lenguaje de Programación.	20
2.3. Framework de desarrollo.	20
2.4. Entorno de Desarrollo Integrado (IDE).....	21
2.5. Sistema de Gestor de Base de Datos.	22
2.6. Quantum GIS.....	22
2.7. Metodología de desarrollo.	23
2.8. Conclusiones parciales.....	25

Capítulo 3	27
Presentación de la solución propuesta.	
3.1. Introducción.....	27
3.2. Fase de Exploración.....	27
3.1.1. Historias de Usuarios.	27
3.3. Fase de Planificación.	32
3.3.1. Estimación de esfuerzos por historia de usuario.	32
3.3.2. Plan de iteraciones.....	33
3.3.3. Plan de duración de las iteraciones.....	34
3.4. Conclusiones Parciales.	35
Capítulo 4	36
Construcción y validación de la solución propuesta.	
4.1. Introducción.	36
4.2. Diseño de la solución propuesta.	36
4.2.1. Tarjetas CRC.	36
4.3. Desarrollo de las iteraciones.....	39
4.3.1. Iteración 1:.....	40
4.3.2. Iteración 2:.....	41
4.3.3. Iteración 3:.....	42
4.3.4. Iteración 4:.....	44
4.3.5. Iteración 5:.....	46
4.4. Requisitos no funcionales del sistema.	48
4.5. Pruebas.	50
4.5.1. Pruebas unitarias.	51
4.5.2. Pruebas de aceptación.	51
4.6. Conclusiones Parciales.....	56
Conclusiones Generales	57
Recomendaciones	58
Referencias Bibliográficas	59
Bibliografía Consultada	61

Índice de Figuras

Fig. 1 Funcionamiento de los SIG.	7
Fig.2 Mapa corocromático.	12
Fig.3 Mapa de coropleta.	13
Fig.4 Mapa de isopleta.	13
Fig.5 Mapa lineal.	14

Tabla #1: Plantilla de Historias de Usuarios	28
Tabla #2: HU Cargar Capa.	28
Tabla #3: HU Visualizar Capa.....	29
Tabla #4: HU Mostrar Capas.	29
Tabla #5: HU Desplazar Mapa.....	29
Tabla #6: HU Aumentar Zoom.	30
Tabla #7: HU Alejar Zoom.....	30
Tabla #8: HU Vista Completa.....	31
Tabla #9: HU Calcular Distancia.	31
Tabla #10: HU Analizar Demanda.....	31
Tabla #11: HU Guardar Imagen.....	32
Tabla #12: Estimación de esfuerzos por historia de usuario.	33
Tabla #13. Plan de duración de las iteraciones.....	34
Tabla #14: Plantilla para Tarjetas CRC.....	37
Tabla #15: ConexionPgSQL.	37
Tabla #16: Visor.....	37
Tabla #17: MapTool.....	38
Tabla #18: Conexión.....	38
Tabla #19: GaSIG.....	38
Tabla #20: Cafeterias.....	39
Tabla #21: Distance.....	39
Tabla #22: Point.....	39
Tabla #23: PointTema.....	39
Tabla #24: HU abordadas en la primera iteración.....	40
Tabla #25: Tarea #1 de la HU #1.....	41
Tabla #26: Tarea #1 de la HU #2.....	41
Tabla #27: Tarea #1 de la HU #3.....	41
Tabla #28: HU abordadas en la segunda iteración.....	42
Tabla #29: Tarea #1 de la HU #4.....	42
Tabla #30: HU abordadas en la tercera iteración.....	42
Tabla #31: Tarea #1 de la HU #5.....	43
Tabla #32: Tarea #1 de la HU #6.....	43
Tabla #33: Tarea #1 de la HU #7.....	44
Tabla #34: HU abordadas en la cuarta iteración.....	44
Tabla #35: Tarea #1 de la HU #8.....	44

Tabla #36: Tarea #2 de la HU #8.....	46
Tabla #37: HU abordadas en la quinta iteración.	47
Tabla #38: Tarea #1 de la HU #9.....	47
Tabla #39: Tarea #1 de la HU #10.....	48
Tabla #40: Prueba de aceptación para la HU “Mostrar Mapa.”.....	52
Tabla #41: Prueba de aceptación para la HU “Desplazar Mapa.”.....	52
Tabla #42: Prueba de aceptación para la HU “Aumentar Zoom.”.....	53
Tabla #43: Prueba de aceptación para la HU “Alejar Zoom.”.....	53
Tabla #44: Prueba de aceptación para la HU “Zoom General.”.....	53
Tabla #45: Prueba de aceptación para la HU “Calcular Distancia.”.....	54
Tabla #46: Prueba de aceptación para la HU “Analizar Demanda.”.....	54
Tabla #47: Prueba de aceptación para la HU “Guardar Imagen.”.....	55
Tabla #48: Resultados de las pruebas de aceptación.....	56

Introducción

Cada día, la informática adquiere más relevancia en la vida de las personas y en las empresas. El desarrollo de las Tecnologías de Información y las Comunicaciones (TICs) ha revolucionado el mundo dando paso a grandes progresos en los distintos sectores de la actividad humana como: medicina, biología, ingeniería, industria, investigación científica, comunicaciones entre otros. Algunos de los ejemplos que se pueden encontrar dentro de estos sectores son: monitorización y control de procesos, robots industriales, control de transportes, diseño computarizado, aplicaciones de multimedia, entre otras variedades de soluciones informáticas que sirven de gran ayuda para la sociedad.

Las características propias de Cuba, un país con escasos recursos minerales y eminentemente agrícola, la obliga a depender de su capacidad de superación en el ámbito profesional para su progreso. Los logros alcanzados en sectores como la medicina, la biotecnología, la educación y el deporte avalan los esfuerzos del país en el impulso de la obtención y aplicación del conocimiento. Cuba apuesta actualmente al desarrollo de la informática como variante de avance tecnológico. Los especialistas de las TICs están llamados a jugar un papel protagónico en el futuro económico y social del país. La máxima dirección del Estado Cubano ha puesto interés especial en la accesibilidad a las tecnologías de la información y las comunicaciones para convertir a Cuba en paradigma de la sociedad de la información y el conocimiento para todos. Una de las principales muestras que evidencian el interés del país de potenciar el uso de las TICs es la creación de la Universidad de las Ciencias Informáticas (UCI).

La UCI surge como "Proyecto Futuro", con dos objetivos fundamentales: informatizar el país y desarrollar la industria del Software. Este centro universitario ha proporcionado beneficios a toda la isla, sus software tienen gran utilidad en la salud, el transporte, la educación, el turismo, el MININT y la FAR. Los productos realizados por la UCI han dejado la huella social de su aplicación en los diferentes sectores de la economía, la amplia gama de soluciones que brindan estos proyectos es precisamente lo que permite este impacto.

Las aplicaciones informáticas pueden ser de gestión, de apoyo a la educación (didácticas), de simulación de fenómenos, de representación espacial entre otros.

La representación espacial es hoy en día una de las ramas de la informática que mayor utilización tiene en las diferentes esferas sociales. El solo hecho de poder

ubicar geográficamente objetos sobre un mapa hace que se convierta en una solución extremadamente útil. Su utilización se considera un instrumento muy potente en la toma de decisiones de cualquier rama de la sociedad. Las aplicaciones que responden a las exigencias de representar objetos espacialmente sobre los mapas se les denominan Sistemas de Información Geográfica (SIG).

El surgimiento de los SIG viene dado desde que el hombre primitivo pintaba en las paredes de las cuevas los animales que cazaban, asociando estos dibujos con trazas lineales. Con el transcurso del tiempo y de acuerdo a las necesidades del hombre estos métodos de comunicación fueron cambiando y desarrollándose cada vez más rápido, hasta que en 1962 se observó claramente la primera utilización real de los SIG en el mundo, con el nombre de Sistema de Información Geográfica de Canadá, este permitía superponer capas de información, realizar mediciones y llevar a cabo digitalizaciones y escaneos de datos. (1)

Los SIG desempeñan un papel fundamental en la representación y análisis de la información geográfica, debido a que proveen los medios necesarios para la captura, organización, manipulación y uso de la información, es decir es una herramienta que permite obtener, almacenar, manipular, desplegar, analizar y modelar escenarios de la realidad geográfica de una manera fácil de comprender, agilizando y fundamentando la toma de decisiones. (1)

En la UCI se han desarrollado aplicaciones informáticas que responden a las exigencias de un SIG. El producto GeneSIG como plataforma para la creación de SIG aplicados a diferentes sectores es el ejemplo más claro de los resultados obtenidos en esta área por la Universidad. Aplicativos de este producto han sido entregados a diferentes organismos como por ejemplo el SIGSalud, solución realizada para el Ministerio de Salud Pública, el SIGONRM al Ministerio de la Industria Básica, específicamente para la Oficina Nacional de Recursos Minerales y una solución que se realizó para la propia comunidad de la UCI, el SIGUCI.

La UCI es una entidad que tiene una gran extensión territorial por lo que se hace muy difícil llevar una correcta distribución de los alimentos a todas las unidades de servicios. Los productos gastronómicos de la UCI se distribuyen actualmente de una manera ineficiente, las cafeterías reciben productos sin tener en cuenta la demanda de las mismas atendiendo a su ubicación geográfica, es decir que no se analiza por los lugares donde el flujo de personas es mayor. Además el gasto innecesario de

combustible en la distribución de los productos es claramente visible, pues no se tienen en cuenta elementos referentes a la ruta más corta entre las cafeterías o entre el centro de elaboración y las cafeterías. Para los directivos de gastronomía en la universidad se hace muy difícil dar solución a lo antes planteado por lo que se define el siguiente **problema a resolver**: ¿Cómo realizar el análisis espacial de la distribución de los productos gastronómicos de la Universidad de las Ciencias Informáticas? Teniendo en cuenta lo planteado se definió como **objeto de estudio** los procesos de análisis espacial de la distribución de los productos gastronómicos estableciendo como **campo de acción** la automatización de los procesos de análisis espacial de la distribución de los productos gastronómicos en la Universidad de las Ciencias Informáticas.

Se define como **objetivo general** de la investigación, desarrollar un Sistema de Información Geográfica que permita realizar el análisis espacial de la distribución de los productos gastronómicos de la Universidad de las Ciencias Informáticas. Para dar cumplimiento al objetivo general se trazaron un conjunto de **objetivos específicos**:

- Definir el marco teórico de la investigación.
- Establecer herramientas para el desarrollo de la aplicación.
- Desarrollar el sistema.

Teniendo como **idea a defender** que si se implementa un software que realice el análisis espacial de la distribución de los productos gastronómicos de la Universidad de las Ciencias Informáticas, permitirá mejorar el área de los servicios gastronómicos en la universidad.

Para dar cumplimiento al objetivo se trazaron las siguientes **tareas de la investigación**:

- Efectuar la fundamentación teórica de la investigación.
- Realizar propuesta de herramientas que permitan desarrollar la solución que se desea.
- Realizar una propuesta de solución.
- Construir la propuesta de solución.
- Realizar las pruebas del sistema.

Como **posibles resultados** se tienen:

- Sistema de Información Geográfica que permita realizar el análisis espacial de la distribución de los productos gastronómicos de la Universidad.
- Documentación técnica del proceso de desarrollo.

Métodos Científicos de Investigación utilizados:

Los métodos teóricos son:

- Analítico-Sintético: Se realizó un estudio con profundidad de toda la información acerca de las tecnologías, metodologías y herramientas posibles a ser utilizadas en el desarrollo del SIG, pudiendo definir con mayor certeza las mismas, sintetizando sus características y analizando la viabilidad de cada una.
- Histórico-Lógico: Para poder realizar este trabajo primeramente se analizó todo referente a las redes de flujo, así como los modelos de redes. Se realizó un estudio de los mapas temáticos y en un tercer plano se vieron los SIG existentes en Cuba y en el resto del mundo.
- Modelación: Se realizaron los modelos correspondientes al ciclo de vida del desarrollo del SIG, esto propició facilidades a la hora de cumplir con las tareas de análisis y diseño de los procesos que intervinieron en la aplicación, así como para la implementación del sistema.
- Inductivo-Deductivo: Este método se aplicó a la hora de inducir y deducir el lenguaje de programación que se escogió, pues son las formas de razonamiento que permiten llegar a un grupo de conocimientos generalizadores, tanto desde el análisis de lo particular a lo general, como desde el análisis de elementos generalizadores a uno de menor nivel de generalización.

El presente trabajo consta de 4 capítulos:

Capítulo 1: Fundamentación Teórica: En este capítulo se describen los conceptos fundamentales asociados al dominio del problema. Se realiza un estudio acerca de las redes lineales, así como los modelos de redes y de los elementos asociados a la distribución de forma genérica y en particular la distribución de alimentos. Finalmente se realiza un análisis de otras soluciones existentes en el mundo.

Capítulo 2. Tecnologías y herramientas a utilizar: En este capítulo se explica el uso de las tecnologías y las herramientas que se utilizaron para desarrollar el SIG.

Capítulo 3. Presentación de la solución propuesta: Este capítulo aborda las dos primeras fases de la Programación Extrema (XP). En la fase de exploración se define las historias de usuario, para un mejor entendimiento del software y en la fase de planificación se realiza la estimación de esfuerzos por historia de usuario.

Capítulo 4. Construcción y validación de la solución propuesta: Este capítulo aborda las restantes fases de la Programación Extrema (XP), construcción y prueba. Donde se describen las tarjetas CRC(Contenido, Responsabilidad y Colaboración) y se detallan las cinco iteraciones llevadas a cabo durante la etapa de construcción de la aplicación, así como las tareas generadas por cada historia de usuario y las pruebas de aceptación efectuadas sobre el sistema.

Capítulo 1

Fundamentación Teórica.

1.1 Introducción.

En este capítulo se expone la fundamentación teórica, que no es más que conceptos asociados al dominio del problema; así como el objeto de estudio donde se pone de manifiesto que son las redes lineales, redes de flujo y modelos de redes. Se abordan temas como la tematización que se utilizará en el software y se realizará un estudio de las soluciones que existen tanto en el mundo como en Cuba.

1.2 Conceptos asociados al dominio del problema.

1.2.1. Sistemas de Información Geográfica.

Los Sistemas de Información Geográfica (SIG) son herramientas necesarias para superar la visión sectorial y consolidar una comprensión integral del territorio, mediante la interacción de las dimensiones ambiental, cultural, económica, social, espacial entre otras. Los SIG desempeñan un papel fundamental en la representación y análisis de la información geográfica, debido a que proveen los medios necesarios para la captura, organización, manipulación y uso de la información. Esto ha permitido que la aplicación de los SIG sea muy diversa. (2)

El profesor David Rhind (1989) ha definido un Sistema de Información Geográfica como un sistema de hardware, software y procedimientos, diseñados para soportar la captura, el manejo, la manipulación, el análisis, el modelado y el despliegue de datos espacialmente referenciados (georreferenciados), para la solución de los problemas complejos del manejo y planeamiento territorial. (20)

Los SIG funcionan con dos tipos diferentes de información geográfica¹: el modelo vector y el modelo ráster (Fig.1). El modelo ráster ha evolucionado para modelar tales características continuas. Una imagen ráster comprende una colección de celdas (píxel) de una grilla más como un mapa o una figura escaneada. Ambos modelos para almacenar datos geográficos tienen ventajas y desventajas únicas y los SIG modernos pueden manejar varios tipos. En el modelo vector, la información sobre puntos, líneas y polígonos se almacena como una colección de coordenadas x, y. La ubicación de una característica puntual, pueden describirse con un sólo punto x, y. Las

¹ Información geográfica: Conjunto de datos espaciales, los cuales brindan una información de algún hecho o fenómeno.

características lineales, pueden almacenarse como un conjunto de puntos de coordenadas x , y . Las características poligonales, pueden almacenarse como un circuito cerrado de coordenadas. El modelo vector es extremadamente útil para describir características discretas, pero menos útil para describir características de variación continua. (3)

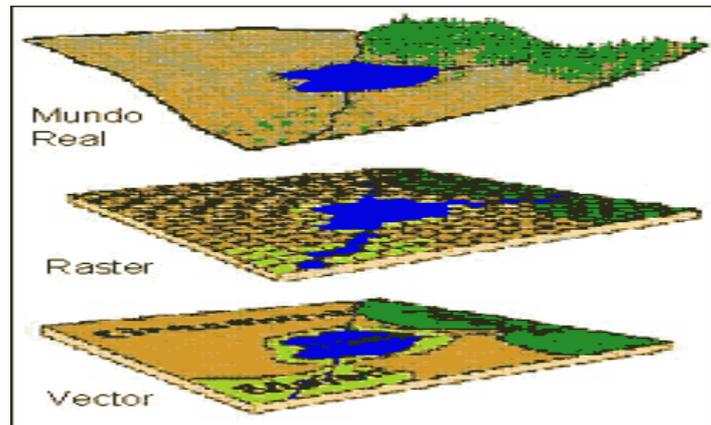


Fig. 1 Funcionamiento de los SIG.

1.2.2. Redes Lineales.

En términos matemáticos, una red es un grafo, donde los nodos representan agentes individuales y los arcos o enlaces representan la transferencia entre agentes de un bien de utilidad, por ejemplo: información, prestigio personal, etc. (21)

Una red lineal queda descrita por un sistema de ecuaciones diferenciales lineales de primer orden. El número de ecuaciones es igual al número de componentes dinámicas, esto si existe un árbol que contenga a todos los condensadores y fuentes independientes de tensión, y que las cuerdas contengan a todos los inductores y fuentes de corriente. (22)

Se puede afirmar que una red lineal es aquella que está formada por la interconexión de componentes elementales lineales, donde los modelos matemáticos pueden emplearse para representar estos componentes.

1.2.2.1. Red de Flujo.

Las redes de flujo son modelos matemáticos aplicables a situaciones tales como: sistemas de tuberías (para fluidos como agua, petróleo o gas), redes de cableado eléctrico, sistemas de carreteras, sistemas de transporte de mercancías, etc.

La definición formal es la siguiente:

Definición: Una red de flujo es un grafo $G = (V; E)$ con una función de capacidad $c: E \rightarrow R^+$ y dos vértices distinguidos, llamados fuente y sumidero. (23)

Una red de flujo es un grafo dirigido, donde la fuente es quien produce o inicia el trasiego de algún material o producto por los arcos, estos últimos, vistos como caminos o conductos. (4)

Se puede alegar que las técnicas de flujo de redes están orientadas a optimizar situaciones vinculadas a todas aquellas que puedan representarse mediante una red, donde los nodos representan las estaciones o las ciudades y los arcos, los caminos, los cables, las tuberías y el flujo lo representan los camiones.

1.2.2.1.1. Modelos de Redes.

Los problemas de optimización de redes se pueden representar en términos generales a través de uno de estos tres modelos:

1. Modelo de minimización de redes (Problema del árbol de mínima expansión).

El modelo de minimización de redes o problema del árbol de mínima expansión tiene que ver con la determinación de los ramales que pueden unir todos los nodos de una red, tal que minimice la suma de las longitudes de los ramales escogidos. No se deben incluir ciclos en la solución del problema.

Para crear el árbol de expansión mínima se tienen las siguientes características:

- Se tienen los nodos de una red pero no las ligaduras². En su lugar se proporcionan las ligaduras potenciales y la longitud positiva para cada una si se inserta en la red. (Las medidas alternativas para la longitud de una ligadura incluyen distancia, costo y tiempo.)
- Se desea diseñar la red con suficientes ligaduras para satisfacer el requisito de que haya un camino entre cada par de nodos.
- El objetivo es satisfacer este requisito de manera que se minimice la longitud total de las ligaduras insertadas en la red.

Una red con n nodos requiere sólo $(n-1)$ ligaduras para proporcionar una trayectoria entre cada par de nodos. Las $(n-1)$ ligaduras deben elegirse de tal manera que la red resultante formen un árbol de expansión. Por tanto el problema es hallar el árbol de expansión con la longitud total mínima de sus ligaduras.

Algoritmo para construir el árbol de expansión mínima: (24)

²Ligaduras: Se refiere a las aristas entre dos nodos.

- Se selecciona, de manera arbitraria, cualquier nodo y se conecta (es decir, se agrega una ligadura entre ellos) al nodo distinto más cercano.
- Se identifica el nodo no conectado más cercano a un nodo conectado y se conectan estos dos nodos (es decir, se agrega una ligadura entre ellos). Este paso se repite hasta que todos los nodos están conectados.
- Empates: los empates para el nodo más cercano (paso 1) o para el nodo no conectado (paso 2) más cercano se pueden romper en forma arbitraria y el algoritmo debe llegar a una solución óptima. No obstante, estos empates son señal de que pueden existir (pero no necesariamente) soluciones óptimas múltiples. Todas esas soluciones se pueden identificar si se trabaja con las demás formas de romper los empates hasta el final.

2. Modelo de Flujo Máximo.

Se trata de enlazar un nodo fuente y un nodo destino a través de una red de arcos dirigidos. Cada arco tiene una capacidad máxima de flujo admisible. El objetivo es el de obtener la máxima capacidad de flujo entre la fuente y el destino.

Características:

- Todo flujo a través de una red conexa dirigida se origina en un nodo, llamado fuente, y termina en otro nodo llamado destino.
- Los nodos restantes son nodos de trasbordo.
- Se permite el flujo a través de un arco sólo en la dirección indicada por la flecha, donde la cantidad máxima de flujo está dada por la capacidad del arco. En la fuente, todos los arcos señalan hacia fuera. En el destino, todos señalan hacia el nodo.
- El objetivo es maximizar la cantidad total de flujo de la fuente al destino. Esta cantidad se mide en cualquiera de las dos maneras equivalentes, esto es, la cantidad que sale de la fuente o la cantidad que entra al destino.

El problema de flujo máximo se puede formular como un problema de programación lineal, se puede resolver con el método simplex y usar cualquier software. Sin embargo, se dispone de un algoritmo de trayectorias aumentadas mucho más eficientes. El algoritmo se basa en dos conceptos intuitivos, el de red residual y el de trayectoria aumentada.

Algoritmo de la trayectoria de aumento para el problema de flujo máximo: (24)

- Se identifica una trayectoria de aumento encontrando alguna trayectoria dirigida del origen al destino en la red residual, tal que cada arco sobre esta trayectoria tiene capacidad residual estrictamente positiva. (Si no existe una, los flujos netos asignados constituyen un patrón del flujo óptimo).
- Se identifica la capacidad residual c de esta trayectoria de aumento encontrando el mínimo de las capacidades residuales de los arcos sobre esta trayectoria. Se aumenta en c el flujo de esta trayectoria.
- Se disminuye en c la capacidad residual de cada arco en esta trayectoria de aumento. Se aumenta en c la capacidad residual de cada arco en la dirección opuesta en esta trayectoria. Se regresa al comienzo del algoritmo.

3. Modelo de la ruta más corta.

Considere una red conexa y no dirigida con dos nodos llamados origen y destino. A cada ligadura se asocia una distancia no negativa. El objetivo es encontrar la ruta más corta (la trayectoria con la mínima distancia total) del origen al destino. Se dispone de un algoritmo bastante sencillo para este problema. La esencia del procedimiento es que analiza toda la red a partir del origen; identifica de manera sucesiva la ruta más corta a cada uno de los nodos en orden ascendente de sus distancias (más cortas), desde el origen; el problema queda resuelto en el momento de llegar al nodo destino.

Algoritmo de la ruta más corta: (24)

- Objetivo de la n -ésima iteración: encontrar el n -ésimo nodo más cercano al origen. (Este paso se repetirá para $n=1,2,\dots$ hasta que el n -ésimo nodo más cercano sea el nodo destino.)
- Datos para la n -ésima iteración: $n-1$ nodos más cercanos al origen (encontrados en las iteraciones previas), incluida su ruta más corta y la distancia desde el origen. (Estos nodos y el origen se llaman nodos resueltos, el resto son nodos no resueltos). Candidatos para el n -ésimo nodo más cercano: Cada nodo resuelto que tiene conexión directa por una ligadura con uno o más nodos no resueltos proporciona un candidato, y éste es el nodo no resuelto que tiene la ligadura más corta. (Los empates proporcionan candidatos adicionales).
- Cálculo del n -ésimo nodo más cercano: para cada nodo resuelto y sus candidatos, se suma la distancia entre ellos y la distancia de la ruta más corta desde el origen a este nodo resuelto. El candidato con la distancia total más pequeña es el n -ésimo nodo más cercano (los empates proporcionan nodos resueltos adicionales), y su ruta más corta es la que genera esta distancia.

Después de haber analizado las redes de flujo y los modelos matemáticos se arribó a la conclusión que el trabajo con las redes no aplica a la solución que se pretende dar, debido a que no se cuenta con un grafo que represente los viales de la UCI donde se pueda emplear este tipo de análisis. No obstante la investigación permitió sentar las bases para lograr una solución más completa a partir de diversos criterios que se adquirieron.

1.2.3. Distribución.

La distribución es la función comercial de poner los productos al alcance del mercado. Existen varios tipos de distribución, entre los que se destacan:

➤ **Distribución exclusiva:**

El intercambio tiene la exclusividad para vender el producto en una zona geográfica determinada.

➤ **Distribución selectiva:**

El fabricante selecciona algunos puntos de venta por el estilo del negocio, por lo que representa, por la instalaciones, entre otros.

➤ **Distribución intensiva:**

El fabricante busca para su producto todos los establecimientos de la misma rama comercial en los que se pueden vender.

➤ **Distribución extensiva:**

El fabricante busca poder vender sus productos en todos los establecimientos posibles de la misma rama comercial y también en los establecimientos que no sean de la misma rama comercial. (26)

Actualmente en la UCI no existe una distribución que se ajuste a algunos de estos tipos de distribución planteados anteriormente. Pero la que se utiliza en estos momentos está más cercana a la distribución selectiva, pues a las cafeterías más céntricas y de más demanda se debería llevar más cantidad de productos y más variedad de ofertas que a las otras.

1.2.4. Mapas Temáticos.

A medida que los mapas se han vuelto más complejos, ha ido aumentando la presentación de subconjuntos de la información original. Estos son los mapas para

finos especiales, de distribución o temáticos. Pueden mostrar la distribución espacial³ de una variedad casi ilimitada de características, y su principal objetivo es la representación gráfica de los patrones y relaciones espaciales, en la medida en que es posible relacionarlos con el espacio geográfico y transformarlos en símbolos cartográficos⁴. (18)

Estos tipos de mapas son de gran importancia para los SIG debido al amplio abanico de temas que abarcan. Una parte de estos mapas se encuentran en el comercio aunque la mayoría se prepara para un fin específico, por ejemplo para una investigación, o ilustrar una zona particular en una publicación, o como ayuda para algunos tipos de proyectos de planificación.

1.2.4.1. Tipos de Mapas Temáticos.

Mapas corocromáticos: Muestran la distribución no cuantitativa de cualquier característica en la superficie de la tierra, es decir, simplemente ilustran la presencia o ausencia de una determinada característica en una zona dada (Fig.2).

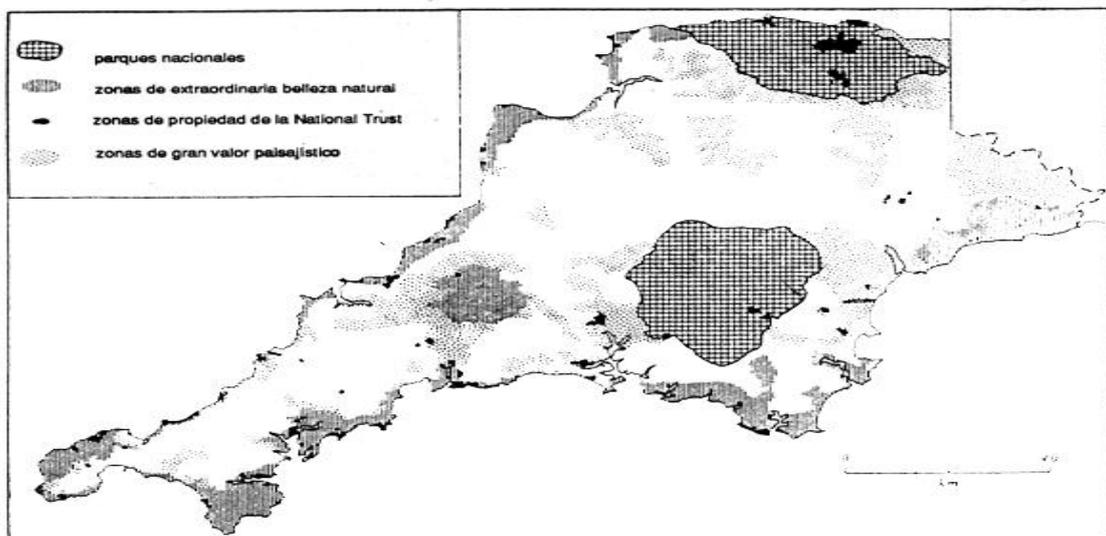


Fig.2 Mapa corocromático.

Mapas de coropletas: Estos mapas denotan los valores medios por unidad de superficie de una región administrativa para la que se dispone de estadísticas, por

³ Distribución espacial: La distribución espacial no es más que la forma en que los objetos se distribuyen en un territorio determinado. La dimensión espacial es un factor de gran importancia en el análisis de las interrelaciones entre objetos y territorios.

⁴ Símbolos cartográficos: Son representaciones gráficas que presentan los diferentes elementos presentes sobre la superficie terrestre.

ejemplo, de la densidad de población o de un rendimiento por unidad de superficie (Fig.3)

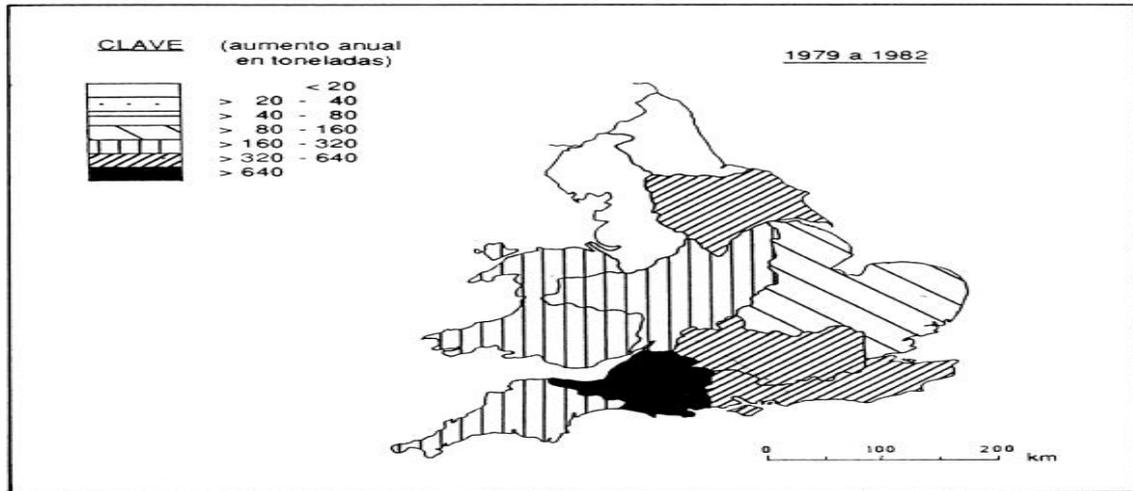


Fig.3 Mapa de coropleta.

Mapas de isopletas: La base de estos mapas es la representación gráfica de los valores de una determinada superficie, que luego se utilizan como puntos para interpolar las isolíneas (o isopletas). Entre las isopletas comunes figuran las curvas de nivel (que indican la altura), las isotermas (las temperaturas), las isobaras (la presión atmosférica) y las isohalinas (la salinidad del mar), pero estas líneas se utilizan también para representar cantidades menos usuales, como las horas de viaje desde un punto nodal de transporte, las variaciones de los costos en una superficie dada o el potencial económico en una gran unidad de tierra (Fig. 4).

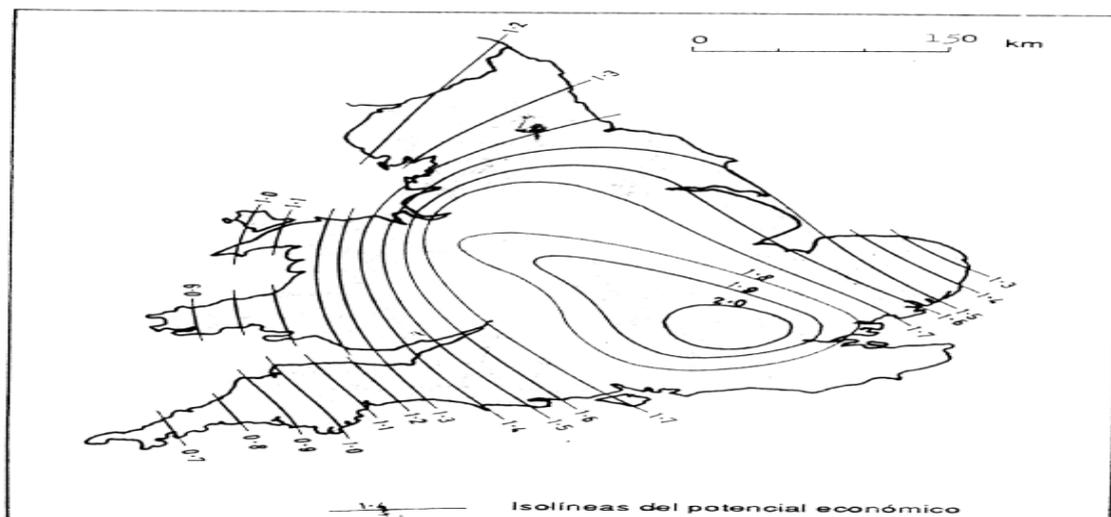


Fig.4 Mapa de isopleta.

Mapas puntiformes: Muchas distribuciones de factores físicos y humanos son puntiformes, es decir, aparecen en un mapa como puntos que pueden indicar pautas de dispersión tendentes hacia una distribución uniforme o aleatoria. Todo aspecto específico de un lugar se puede representar en un mapa puntiforme.

Mapas lineales: Los mapas lineales pueden representar simplemente la ruta o trazado efectivo de una característica, o bien ser volumétricos, es decir, indicar las líneas de movimiento de manera que su espesor sea proporcional al volumen de los bienes o del tráfico que circula por esas rutas. Aunque hay diversas características que presentan distribución lineal, las más conocidas son algunas características físicas, como los ríos, o humanas (asociadas generalmente con las comunicaciones), como los caminos, vías férreas, rutas aéreas, tuberías, líneas de transmisión de energía eléctrica, etc. (Fig. 5)

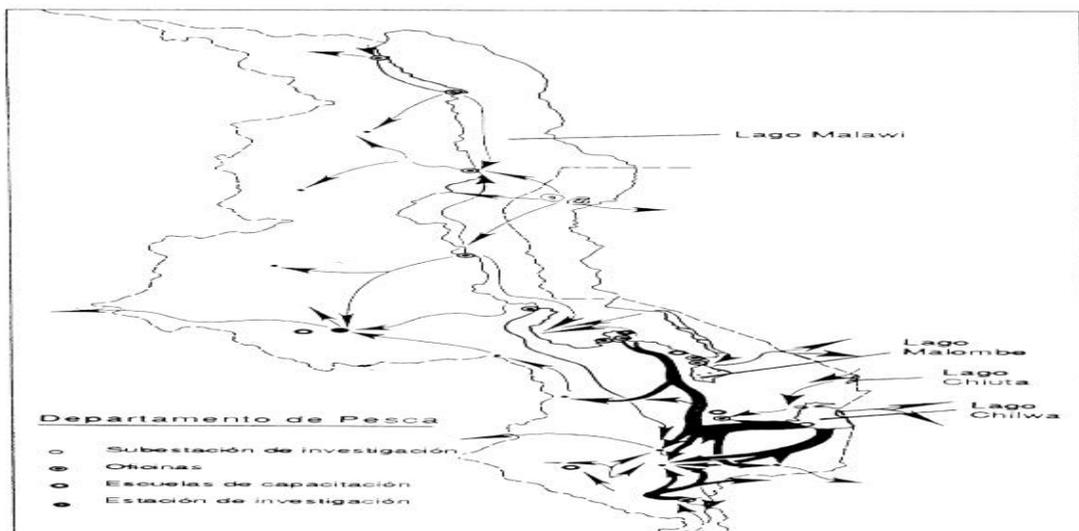


Fig.5 Mapa lineal.

El mapa que se ajustó a la hora de conocer las cafeterías que más demanda tenían fue el de coropletas, pues estos mapas denotan los valores medios por unidad de superficie de una región. El mapa lineal fue otro que se utilizó debido a que pueden representar simplemente la ruta o trazado efectivo de una característica como los caminos entre una cafetería u otra.

1.3. Análisis de otras soluciones existentes.

En la actualidad existen muchas soluciones de SIG, algunas de estas soluciones serían:

1.3.1. Quantum GIS.

Quantum GIS (QGIS) es un SIG de código libre, con grandes potencialidades para la edición de mapas, para plataformas GNU/Linux, Unix, Mac OS y Microsoft Windows. Permite manejar formatos ráster y vectoriales, así como bases de datos.

Características:

QGIS tiene muchas funciones y características comunes a todos los SIG. Las características son divididas en elementos del núcleo y complementos. (6)

Elementos del núcleo:

- Soporte ráster y vectorial.
- Soporte para PostgreSQL con tablas espaciales utilizando PostGIS.
- Integración con GRASS⁵, incluida visualización, edición y análisis.
- Diseño de Mapas.
- Panel de Vista General.
- Guardar y recuperar proyectos.
- Exportar ficheros map a Mapserver.
- Cambiar simbología vectorial y ráster.
- Arquitectura extensible con complementos.

Complementos:

- Añadir capas WFS.
- Añadir capas de texto delimitado.
- Decoración (etiqueta de copyright, flecha de Norte y barra de escala)
- Georreferenciación.
- Herramientas GPS.
- GRASS. (6)

1.3.2. ArcGIS.

El ArcGIS Desktop es como un conjunto de aplicaciones integradas: ArcMap⁶, ArcCatalog⁷ y ArcToolbox⁸, para poder construir un SIG completo en su organización,

⁵ GRASS: Es un SIG con capacidad ráster y vectorial.

⁶ ArcMap: Es un software de Sistema de Información Geográfica (SIG) creado por ESRI para mapeo digital.

⁷ ArcCatalog: Es un software de Sistema de Información Geográfica (SIG) creado por ESRI diseñado para explorar y administrar los datos.

los cuales posibilitan el análisis espacial, la administración de datos y mapeo. Este fue producido y comercializado por ESRI, el mismo se agrupa en varias aplicaciones para la captura, edición, manejo, análisis, tratamiento, diseño, publicación e impresión de información geográfica. El ArcGIS desktop es un sistema amplio, integrado, escalable, diseñado para satisfacer las necesidades de un amplio rango de usuarios. (7)

1.3.3. ArcView.

Es un software SIG para visualizar, crear, manipular y gestionar información geográfica, estos corresponden a lugares, direcciones, posiciones en terreno, áreas urbanas y rurales; regiones y cualquier tipo de ubicaciones en terrenos determinados. ArcView GIS es el software de escritorio más usado de los SIG del mundo, porque entrega de una forma fácil el trabajo en datos geográficos. Tiene una interfaz gráfica amigable, en la cual se puede desplegar de manera rápida la información geográfica. El aprendizaje del software es rápido, teniendo algunos conocimientos de Sistemas de Información Geográfica previos. (7)

Algunas funcionalidades de son: (8)

- Creación de mapas.
- Consultas espaciales.
- Análisis y modelamiento básicos.
- Edición simple de elementos.
- Integración de datos.

1.3.4. ArcEditor.

ArcEditor es parte de la familia ArcGIS, con esta modalidad puedes editar, manipular y gestionar datos de naturaleza geográfica. Tiene toda la funcionalidad de ArcView GIS agregando más herramientas que puedan crear, editar y asegurar el proceso de explotación de datos con una calidad a toda prueba. Está hecho para soportar la edición desde múltiples editores como de un sólo editor de datos geográficos. Tiene amplio y variado conjunto de herramientas para trabajar la gráfica y los datos alfanuméricos. Este se utiliza para trabajar con información compleja. El amplio número de herramientas y los diferentes procesos de trabajo con que cuenta permiten generar y mantener la información geográfica de manera sencilla y eficiente. (7)

⁸ArcToolbox: Es un software de Sistema de Información Geográfico (SIG) creado por ESRI para usar en el geoprocetamiento de datos.

Algunas funcionalidades son:

- Automatizar el control de calidad.
- Diseñar mejor bases de datos geográficas.
- Crear y editar elementos de topología.
- Diseñar planes de edición, validación y guardar la información tratada.
- Mejorar el rendimiento de la edición.

1.3.5. ArcInfo.

Es el software de SIG de escritorio más completo de la familia de productos de ArcGIS Desktop. Incluye la funcionalidad de ArcEditor y ArcView y adicionalmente análisis espacial avanzado, manipulación de datos extensiva y herramientas de cartografía de alto nivel. Las organizaciones usan el poder de ArcInfo para crear, editar, analizar sus datos para tomar mejores y más rápidas decisiones. (8)

Algunos beneficios:

- Realizar análisis de datos SIG y modelamiento avanzado.
- Tomar ventaja de herramientas diseñadas para análisis de superposición, análisis de proximidad, análisis de superficies y procesamiento ráster.
- Publicar y convertir datos desde y hacia muchos formatos.
- Generar mapas profesionales listos para publicación usando herramientas cartográficas de alto nivel.
- Diseñar símbolos personalizados y ubicar anotaciones y etiquetas sofisticadas en el mapa.

1.3.6. GRASS.

GRASS es un SIG con capacidad ráster y vectorial. Cuenta con sistemas integrados de visualización de datos y de procesado de imagen. Este incluye más de 350 módulos para gestión, procesado, análisis y visualización de datos georreferenciados. GRASS es GNU GPL desde 1999. Esta licencia protege a los contribuidores de GRASS frente al uso de sus contribuciones en proyectos propietarios que no permiten el libre acceso a su código fuente. Este es un SIG modular con datos organizados como coberturas ráster, vectoriales y de puntos, proporciona una gran variedad de herramientas que permiten clasificarlo como un SIG de altas prestaciones. (25)

Principales funcionalidades:

- Integración de datos espaciales.
- Procesado de datos ráster.
- Procesado de datos vectoriales.

- Procesado de datos puntuales.
- Procesado de imágenes.
- Visualización.
- Modelado y simulaciones.
- Soporte para datos temporales.
- Procesado de datos 3D.
- Enlaces con otras herramientas.

Para realizar la investigación la herramienta que se utilizó fue QGIS, por ser esta una aplicación desarrollada por la comunidad de software libre, permitió extraer de su código fuente las bibliotecas más importantes para adaptarlas al negocio específico a desarrollar, dándole solución a la nueva aplicación. QGIS implementa varias funcionalidades que apoyan sobremanera el proceso que se desea modelar, los temas relacionados con la tematización pueden ser utilizados para representar de diversas maneras el estado de un área o cafetería en específico. Por otra parte la gran escalabilidad que presenta la actual arquitectura de esta herramienta permite incorporar nuevas funcionalidades a partir de desarrollo de Plugins⁹ garantizando así mejores prestaciones de la aplicación a partir de nuevas exigencias de los clientes. La familia de ArcGis al ser de software propietario no cede su código fuente, y no se pudo adaptar al negocio específico a desarrollar, por lo que quedó descartada la posibilidad de su utilización. GRASS no se pudo utilizar porque es una aplicación fundamentalmente para realizar análisis complejos, esta herramienta está muy limitada en cuanto a usabilidad para los usuarios, prácticamente todas sus funciones tienen que ser ejecutadas a base de comandos. Esta deficiencia hace de GRASS un software que sea utilizado solo por un pequeño grupo de especialistas, cartógrafos y geólogos que realizan análisis muy complejos y específicos. Muchos expertos consideran a QGIS la interfaz gráfica de GRASS debido a que este utiliza todas las funcionalidades de análisis de esta herramienta.

1.4. Conclusiones Parciales.

En este capítulo se abordaron diversos conceptos asociados al problema para lograr un mejor entendimiento de la investigación. Se realizó un estudio de las redes lineales, redes de flujo y de los modelos de redes, permitiendo ampliar el conocimiento sobre la distancia de un origen a un destino. La tematización fue otro elemento importante, debido a que se comprendieron los diferentes tipos que existían, dando la posibilidad

⁹ Plugins: Es un programa que incrementa o aumenta las funcionalidades de un programa principal.

Capítulo 1: Fundamentación Teórica

de poder utilizar el más adecuado en el sistema. Se estudiaron detalladamente algunas soluciones existentes con el objetivo de poder definir una, para adaptarla al negocio específico a desarrollar.

Capítulo 2

Tecnologías y herramientas a utilizar.

2.1. Introducción.

En el desarrollo de este capítulo, se explica mediante una descripción detallada el uso de las herramientas y tecnologías a utilizar en el desarrollo del software. Se realiza un estudio profundo de la metodología que servirá de gran apoyo para el progreso de la aplicación.

2.2. Lenguaje de Programación.

Un lenguaje de programación es un idioma artificial diseñado para expresar computaciones que pueden ser llevadas a cabo por máquinas como las computadoras. Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.

Para el desarrollo de la aplicación se eligió el lenguaje de programación Python, este proporciona capacidades de Programación Orientada a Objetos (P.O.O) con el fin de incrementar la productividad, calidad y reutilización del software. Se trata de un lenguaje interpretado o de script, con tipado dinámico, fuertemente tipado, con una sintaxis muy limpia y que favorece un código legible.

El intérprete de Python está disponible en multitud de plataformas (UNIX, Solaris, Linux, DOS, Windows, OS/2, Mac OS, etc.) por lo que si no utilizamos bibliotecas específicas de cada plataforma nuestro programa podrá correr en todos estos sistemas sin grandes cambios. (9)

Se llegó a la conclusión de seleccionar este lenguaje debido a que puede acceder a las bibliotecas de QGis, aún siendo estas desarrolladas en el lenguaje de C++. A la hora de implementar la aplicación se tuvo en cuenta que Python además de incrementar la productividad, calidad y reutilización del software es uno de los lenguajes más fáciles de utilizar, por lo que no atrasaría el proceso de desarrollo del software.

2.3. Framework de desarrollo.

Un framework es una estructura de soporte bien definida en la cual otro proyecto de software puede ser organizado y desarrollado. Para el progreso de la interfaz gráfica de usuario de este software se eligió la biblioteca multiplataforma Qt. Esta biblioteca a pesar de ser creada inicialmente para el lenguaje C++, por su fortaleza en apariencia y portabilidad se fueron extendiendo hacia otros lenguajes de programación , ejemplo en

Python esta biblioteca es conocida como PyQt. Esta es concebida como multiplataforma, permitiendo escribir código que se compilará y ejecutará en distintas plataformas, incluyendo Unix, Linux y Windows.

Algunas de sus características que hacen que se llegue a la conclusión de utilizar esta biblioteca son: (9)

- Compatibilidad multiplataforma con un sólo código fuente.
- Disponibilidad del código fuente.
- Excelente documentación.
- Fácilmente internacionabilizable.

2.4. Entorno de Desarrollo Integrado (IDE).

Se seleccionó como entorno de desarrollo Eclipse debido a que facilita enormemente las tareas de edición, compilación y ejecución de programas durante su fase de desarrollo. Además es una aplicación gratuita y de código abierto, disponible en la red para su descarga e incluida ya en muchas distribuciones de Linux.

En sí mismo Eclipse es un marco y un conjunto de servicios para construir un entorno de desarrollo a partir de componentes conectados (plugins). Dentro de estos plugins se encuentra eclipse pydev, el cual permite hacer configuraciones al eclipse para programar en Python, siendo este uno de los principales motivos por el cual se utilizó eclipse, conjuntamente con las siguientes ventajas que ofrece: (10)

- Permite Acceso a todos los ficheros del proyecto.
- Ofrece una Ventana de depuración y errores.
- Información de los parámetros que usa una función.
- Subrayado de errores automáticos.
- Autocompletado.
- Ventana con un completo listado de todas las variables, funciones, objetos, métodos, del fichero que estemos editando.
- Acceso rápido a varios proyectos a la vez.

Después de haber estudiado las ventajas principales de Eclipse se llegó a la conclusión de que era necesaria su utilización para el desarrollo de la aplicación, pues este entorno de desarrollo incluye una serie de características únicas como la refactorización de código, el código de actualizaciones automáticas e instalaciones. Permite a los desarrolladores crear aplicaciones de escritorio para diversos sistemas

operativos, por lo que servía de mucha ayuda para realizar el software, siendo este de escritorio.

2.5. Sistema de Gestor de Base de Datos.

El Sistema de Gestor de Base de Datos que se utilizó fue PostgreSQL, pues es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD¹⁰ y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado.

Se decidió el trabajo con el gestor de bases de datos Postgres SQL, por la potencia que este nos brinda al incorporar los siguientes cuatro conceptos básicos en una vía en la que los usuarios pueden extender fácilmente el sistema, clases, herencia tipos y funciones. Además de otras características que aportan flexibilidad, como son: (12)

- Restricciones.
- Disparadores.
- Reglas.
- Integridad transaccional.

Se llegó a la conclusión de que las características más importantes son la estabilidad, la potencialidad, la robustez y la facilidad de administración e implementación de estándares. Funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. Está disponible para Linux y UNIX en todas sus variantes y Windows 32/64bit. Para el trabajo con datos geográficos existe una extensión llamada PostGIS que contiene una gran cantidad de funciones implementadas. Esta extensión transforma al Sistema de Gestor de Base de Datos PostgreSQL en una base de datos espaciales.

2.6. Quantum GIS.

QGIS es desarrollado en C++, es un visor de datos SIG, Soporta plugins, entre los cuales cabe destacar el de georreferenciación de imágenes ráster. Con esta función de georreferenciación se puede crear archivos para cualquier mapa escaneado o fuente de datos a registrar, aplicando una transformación lineal o mejor aún, una transformación bidimensional. Es multiplataforma, funcionando en Linux, Unix, Mac OSX, y por supuesto en Windows 2000 y XP. Es interesante también destacar que dispone de un amplio abanico de formatos de lectura, ya que utiliza la conocida

¹⁰BSD: Es una licencia de software libre, también permite el uso del código fuente en software no libre.

biblioteca de Datos Geoespaciales de Abstracción. Es también importante enfatizar que soporta funciones de proyección muy dignas para un visualizador de estas características. (14)

En esta investigación se utilizaron las bibliotecas de QGIS para el desarrollo de la aplicación, pues estas proporcionaron el desplazamiento del mapa, el aumento de la escala, así como permitir cargar las capas entre otras funcionalidades que fueron de gran ayuda a la hora de realizar el software.

2.7. Metodología de desarrollo del software.

Para el desarrollo del software se necesitó de una metodología que fuera capaz de cumplir con una serie de requisitos que facilitaran el progreso de la aplicación, para esto se eligió Programación Extrema (XP). Esta es una metodología para el desarrollo de software y consiste básicamente en ajustarse estrictamente a una serie de reglas que se centran en las necesidades del cliente para lograr un producto de buena calidad en poco tiempo. La Programación Extrema es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo del software. Este tipo de método se basa en una realimentación continuada entre el cliente y el equipo de desarrollo con una comunicación fluida entre todos los participantes, también busca la simplicidad en las soluciones implementadas y el coraje para enfrentar los cambios. Este tipo de programación es la adecuada para los proyectos con requisitos imprecisos, muy cambiantes y con un riesgo técnico excesivo.

Básicamente, la metodología XP se basa en cinco valores: (15)

- **Comunicación:** La comunicación permanente es fundamental en XP. Dado que la documentación es escasa, el diálogo frontal, cara a cara, entre desarrolladores, gerentes y el cliente es el medio básico de comunicación. Una buena comunicación tiene que estar presente durante todo el proyecto.
- **Simplicidad:** La sencillez es esencial para que todos puedan entender el código, y se trata de mejorar mediante recodificaciones continuas.
- **Feedback:** Básicamente el continuo contacto con el usuario, al irle entregando las sucesivas versiones, en funcionamiento del producto, permite que este nos dé su valoración y nos comunique, cada vez mejor, lo que realmente quiere en el producto.
- **Coraje:** Básicamente es trabajar muy duro durante las horas dedicadas a ello.

- **Respeto:** Si los miembros de un equipo no se preocupan por sí mismos y por su trabajo, la metodología no puede funcionar. Es necesario ser respetuoso con sus colegas, sus contribuciones, su organización y con las personas cuya vida se toca por el sistema que está escribiendo.

El ciclo de vida ideal consta de 4 fases: (15)

1. **Exploración:** En esta fase, los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo.
2. **Planificación:** En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Esta fase incluye varias iteraciones sobre el sistema antes de ser entregado. El Plan de Entrega está compuesto por iteraciones de no más de tres semanas.
3. **Construcción:** La fase de construcción requiere de las tarjetas CRC (Contenido, Responsabilidad, Colaboración), estas permiten desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos. En esta fase se realiza el desarrollo de las iteraciones, donde las historias de usuario se descomponen en tareas de programación o en tareas de ingeniería.
4. **Prueba:** Esta fase garantiza el correcto funcionamiento del código que se va implementando. Las pruebas se dividen en dos: en pruebas unitarias y pruebas de aceptación. Las unitarias verifican que el código correspondiente a un módulo específico se comporte de manera esperada.

Características fundamentales de la metodología: (16)

- Desarrollo iterativo e incremental: pequeñas mejoras, unas tras otras.
- Pruebas unitarias continuas, frecuentemente repetidas y automatizadas, incluyendo pruebas de regresión. Se aconseja escribir el código de la prueba antes de la codificación.
- Frecuente interacción del equipo de programación con el cliente o usuario. Se recomienda que un representante del cliente trabaje junto al equipo de desarrollo.

- Corrección de todos los errores antes de añadir nueva funcionalidad. Hacer entregas frecuentes.
- Refactorización del código, es decir, reescribir ciertas partes del código para aumentar su legibilidad y mantenibilidad pero sin modificar su comportamiento. Las pruebas han de garantizar que en la refactorización no se ha introducido ningún fallo.
- Propiedad del código compartida: en vez de dividir la responsabilidad en el desarrollo de cada modulo en grupos de trabajo distintos, este método promueve el que todo el personal pueda
- Corregir y extender cualquier parte del proyecto. Las frecuentes pruebas de regresión garantizan que los posibles errores serán detectados.
- Simplicidad en el código: es la mejor manera de que las cosas funcionen. Cuando todo funcione se podrá añadir funcionalidad si es necesario. La programación extrema apuesta que es más sencillo hacer algo simple y tener un poco de trabajo extra para cambiarlo si se requiere, que realizar algo complicado y quizás nunca utilizarlo.

Fundamentalmente se seleccionó esta metodología, porque responde a las necesidades de entregar el software en un corto período de tiempo y porque no se cuenta con un equipo de desarrollo para generar todos los artefactos que pueda generar otra metodología, siendo XP la que se ajusta a que se logre con eficiencia el desarrollo del sistema. Utilizarla tiene como ventaja que el cliente se encuentra dentro del equipo de desarrollo, es decir trabajan en conjunto permitiendo la retroalimentación o una mejor comunicación entre el cliente y el programador.

Otros de los motivos de su uso es que una de las prácticas más importante de XP y que está muy ligada al proceso de formación actual de la universidad es la de Paso Sostenible. Aquí se plantea que en la semana se debe trabajar solo 40 horas, lo que permite que el proyecto se adapte a las horas de producción que se necesitan tener por cada estudiante de la UCI. También se consigue tener un proceso de desarrollo motivado, debido a que XP no permite excesos de trabajos y se logra una mayor integración entre los miembros por la comunicación.

2.8. Conclusiones parciales.

En este capítulo se realizó un estudio profundo de las herramientas, el lenguaje y la metodología que se utilizaron para darle cumplimiento al desarrollo de la aplicación. Adoptando el uso del lenguaje de programación Python para desarrollar todas las funcionalidades de la herramienta. Se escogió a Eclipse como entorno de desarrollo

Capítulo 2: Tecnologías y Herramientas

integrado, debido a que posee un plugin llamado Eclipse_Pydev para la vinculación de este con el lenguaje seleccionado. Se optó como sistema gestor de base de datos a PostgreSQL con su extensión PostGIS, permitiendo el manejo con objetos espaciales. Como metodología utilizada para guiar el proceso de desarrollo del software se identificó a XP, contribuyendo a las necesidades de entregar el proyecto en un corto período de tiempo.

Capítulo 3

Presentación de la solución propuesta.

3.1. Introducción.

En este capítulo se hace una valoración de las dos primeras fases del ciclo de vida de la metodología XP, exploración y planificación. En el que se plasman las historias de usuarios para lograr un mejor entendimiento acerca del software. Estas son escritas por los propios clientes tal y como ven ellos las necesidades del sistema.

3.2. Fase de Exploración.

La primera fase de la metodología de desarrollo XP es exploración, en esta fase los clientes plantean a grandes rasgos las historias de usuario que son de interés para la primera entrega del producto. Asimismo el equipo de trabajo se familiariza con la tecnología, las herramientas y practicas a utilizar en el proyecto. La fase de exploración toma de pocas semanas a pocos meses, dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología. Finalmente se prueba la tecnología y se exploran las posibilidades de la arquitectura del sistema construyendo un prototipo. Es la fase en la que se define el alcance general del proyecto. (19)

3.1.1. Historias de Usuarios.

Las Historias de Usuario (HU) son utilizadas en la metodología de desarrollo ágil XP para representar una breve descripción del comportamiento del sistema. Se trata de tarjetas de papel en las cuales el cliente describe brevemente las características que el sistema debe poseer, sean requisitos funcionales o no funcionales. Emplea terminología del cliente sin lenguaje técnico, se realiza una por cada funcionalidad del sistema, se emplean para hacer estimaciones de tiempo y para el plan de lanzamientos, reemplazan un gran documento de requisitos y presiden la creación de las pruebas de aceptación. Cuando llega la hora de implementar una historia de usuario, el cliente y los desarrolladores se reúnen para concretar y detallar lo que tiene que hacer dicha historia. El tiempo de desarrollo ideal para una historia de usuario es entre 1 y 3 semanas, estas permiten responder rápidamente a los requerimientos cambiantes. (19)

Historias de Usuarios	
Número: Número sucesivo a partir de 1.	Nombre: Identifica la historia de usuario en cuestión.

Capítulo 3: Presentación de la solución

Usuario: Usuarios que interactúan con la HU	
Prioridad en Negocio: Define la relevancia e impacto de la historia de usuario para el negocio de acuerdo con las necesidades del usuario (Alta / Media / Baja).	Riesgo en Desarrollo: Define la dificultad técnica desde el punto de vista del programador (Alto / Medio / Bajo).
Puntos de Estimación: La estimación de esfuerzo se expresa utilizando como medida el punto. Un punto se considera como una semana ideal de trabajo.	Iteración Asignada: Precisa la iteración a la que pertenece la historia de usuarios.
Descripción: Explica en qué consiste la historia de usuario, teniendo en cuenta las acciones realizadas por el usuario y la respuesta brindada por el sistema.	
Observaciones: Información extra que se estime agregar para hacer más comprensible la historia de usuario.	

Tabla #1: Plantilla de Historias de Usuarios

Se identificaron las siguientes Historias de Usuario como parte del proceso de trabajo dentro de la fase de exploración:

Historias de Usuarios	
Número: 1	Nombre: Cargar Capas.
Usuario: Cliente.	
Prioridad en Negocio: Alta.	Riesgo en Desarrollo: Medio.
Puntos de Estimación: 1.	Iteración Asignada: 1.
Descripción: Permite al usuario visualizar todas las capas del mapa.	
Observaciones: -	

Tabla #2: HU Cargar Capa.

Historias de Usuarios	
Número: 2	Nombre: Visualizar Capas.
Usuario: Cliente.	

Capítulo 3: Presentación de la solución

Prioridad en Negocio: Alta.	Riesgo en Desarrollo: Alto.
Puntos de Estimación: 1.	Iteración Asignada: 1.
Descripción: Permite al usuario visualizar todas las capas.	
Observaciones: Debe haberse cargado al menos una capa para permitir el manejo sobre ella.	

Tabla #3: HU Visualizar Capa.

Historias de Usuarios	
Número: 3	Nombre: Mostrar Capas.
Usuario: Cliente.	
Prioridad en Negocio: Alta.	Riesgo en Desarrollo: Medio.
Puntos de Estimación: 1.	Iteración Asignada: 1.
Descripción: Mostrar Capas le permite al usuario escoger la capa que desea mostrar, logrando una mejor visión del mapa.	
Observaciones: La aplicación debe tener varias capas cargadas, para poder seleccionar la que se desee.	

Tabla #4: HU Mostrar Capas.

Historias de Usuarios	
Número: 4	Nombre: Desplazar Mapa.
Usuario: Cliente.	
Prioridad en Negocio: Alta.	Riesgo en Desarrollo: Medio.
Puntos de Estimación: 2.	Iteración Asignada: 2.
Descripción: Desplazar Mapa le permite al usuario mover todo el lienzo o área de trabajo según sea necesario y cómodo para él.	
Observaciones: La aplicación debe tener una capa cargada para poder realizar el desplazamiento en ella.	

Tabla #5: HU Desplazar Mapa.

Capítulo 3: Presentación de la solución

Historias de Usuarios	
Número: 5.	Nombre: Aumentar Zoom.
Usuario: Cliente.	
Prioridad en Negocio: Media.	Riesgo en Desarrollo: Medio.
Puntos de Estimación: 1.	Iteración Asignada: 3.
Descripción: Aumentar Zoom le permite al usuario acercar la imagen, logrando una mejor visión del elemento seleccionado.	
Observaciones: La aplicación debe tener el mapa visualizado para poder permitir el trabajo sobre el mismo.	

Tabla #6: HU Aumentar Zoom.

Historias de Usuarios	
Número: 6.	Nombre: Alejar Zoom.
Usuario: Cliente.	
Prioridad en Negocio: Media.	Riesgo en Desarrollo: Medio.
Puntos de Estimación: 1.	Iteración Asignada: 3.
Descripción: Alejar Zoom le permite al usuario alejar la imagen, logrando una mejor visión del elemento seleccionado.	
Observaciones: La aplicación debe tener el mapa visualizado para poder permitir el trabajo sobre el mismo.	

Tabla #7: HU Alejar Zoom.

Historias de Usuarios	
Número: 7.	Nombre: Vista Completa.
Usuario: Cliente.	
Prioridad en Negocio: Media.	Riesgo en Desarrollo: Medio.
Puntos de Estimación: 1.	Iteración Asignada: 3.
Descripción: Vista Completa le permite al usuario volver a ver el mapa completo sobre el lienzo.	
Observaciones: La aplicación debe tener el mapa visualizado para poder permitir el	

Capítulo 3: Presentación de la solución

trabajo sobre el mismo.

Tabla #8: HU Vista Completa.

Historias de Usuarios	
Número: 8.	Nombre: Calcular Distancia.
Usuario: Cliente.	
Prioridad en Negocio: Alta.	Riesgo en Desarrollo: Alto.
Puntos de Estimación: 3.	Iteración Asignada: 4.
Descripción: Calcular Distancia le permite al usuario conocer la distancia de todas las cafetería.	
Observaciones: La aplicación debe tener el mapa visualizado para poder permitir el trabajo sobre el mismo.	

Tabla #9: HU Calcular Distancia.

Historias de Usuarios	
Número: 9.	Nombre: Analizar Demanda.
Usuario: Cliente.	
Prioridad en Negocio: Alto.	Riesgo en Desarrollo: Alto.
Puntos de Estimación: 2.	Iteración Asignada: 5.
Descripción: La demanda le permite al usuario conocer la cafetería que más demanda tiene, para poder suministrarle el producto.	
Observaciones: La aplicación debe tener el mapa visualizado para poder permitir el trabajo sobre el mismo.	

Tabla #10: HU Analizar Demanda.

Historias de Usuarios	
Número: 10.	Nombre: Guardar Imagen.
Usuario: Cliente.	
Prioridad en Negocio: Baja.	Riesgo en Desarrollo: Medio.
Puntos de Estimación: 1.	Iteración Asignada: 5.

Descripción: Guardar Imagen le permite al usuario archivar la imagen que desee, para uso conveniente.

Observaciones: La aplicación debe tener el mapa visualizado para poder permitir el trabajo sobre el mismo.

Tabla #11: HU Guardar Imagen.

3.3. Fase de Planificación.

En esta fase el cliente establece la prioridad de cada historia de usuario, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario de cada una de ellas. Se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Una entrega debería obtenerse en no más de tres meses. Esta fase dura unos pocos días. Las estimaciones de esfuerzo asociado a la implementación de las historias la establecen los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos. Por otra parte, el equipo de desarrollo mantiene un registro de la “velocidad” de desarrollo, establecida en puntos por iteración, basándose principalmente en la suma de puntos correspondientes a las historias de usuario que fueron terminadas en la última iteración. La planificación se puede realizar basándose en el tiempo o el alcance. La velocidad del proyecto es utilizada para establecer cuántas historias se pueden implementar antes de una fecha determinada o cuánto tiempo tomará implementar un conjunto de historias. Al planificar por tiempo, se multiplica el número de iteraciones por la velocidad del proyecto, determinándose cuántos puntos se pueden completar. Al planificar según alcance del sistema, se divide la suma de puntos de las historias de usuario seleccionadas entre la velocidad del proyecto, obteniendo el número de iteraciones necesarias para su implementación. (19)

3.3.1. Estimación de esfuerzos por historia de usuario.

Se realizó una estimación para cada una de las historias de usuario identificadas, llegando a los resultados que se encuentran a continuación:

Historias de Usuarios	Puntos de Estimación
Cargar Capas.	1 semana
Visualizar Capas.	1 semana

Mostrar Capas.	1 semana
Desplazar Mapa.	2 semana
Aumentar Zoom.	1 semana
Alejar Zoom.	1 semana
Vista Completa.	1 semana
Calcular Distancia.	3 semana
Analizar Demanda.	2 semana
Guardar Imagen.	1 semana

Tabla #12: Estimación de esfuerzos por historia de usuario.

3.3.2. Plan de iteraciones.

Esta fase incluye varias iteraciones sobre el sistema. El Plan de entrega está compuesto por iteraciones de no más de tres semanas. En la primera iteración se puede intentar establecer una arquitectura del sistema que pueda ser utilizada durante el resto del proyecto. Esto se logra escogiendo las historias que fueren la creación de esta arquitectura, sin embargo, esto no siempre es posible ya que es el cliente quien decide qué historias se implementarán en cada iteración. Al final de la última iteración el sistema estará listo para entrar en producción.

Después de haber definido las HU y estimado el esfuerzo propuesto para la realización de cada una de ellas, se tomó la decisión de realizar el sistema en 5 iteraciones, las cuales se detallan a continuación:

Iteración 1:

Esta iteración tiene como objetivo darle cumplimiento a las HU que se consideraron de mayor importancia para el desarrollo del software. Al concluir dicha iteración se contará con todas las funcionalidades descritas en las HU #1, #2 y #3 las cuales hacen alusión a todo lo referente con cargar, visualizar y mostrar las capas.

Iteración 2:

Esta iteración tiene como finalidad desarrollar la HU #4. Esta historia es la encargada de desplazar el mapa, esto le permite al usuario mover todo el lienzo o área de trabajo según le sea necesario. La versión que se obtendrá de esta iteración en unión con la entregada en la iteración anterior se le dará al cliente para verificar si cumple con las necesidades antes acordadas con él.

Capítulo 3: Presentación de la solución

Iteración 3:

En esta iteración se dará cumplimiento a las HU que corresponden a los números #4, #5 y #6. Estas son referentes al zoom del mapa, le permiten al cliente acercar y alejar la parte seleccionada del mapa, mientras que la vista completa le brinda la posibilidad de volver a ver el mapa completo, siéndole más cómodo navegar por todo el lienzo.

Iteración 4:

Esta iteración tiene como propósito desarrollar la HU #7. Esta historia es la que se encarga de calcular la distancia entre las cafeterías, de esta manera el usuario puede elegir el camino más corto para llegar al destino.

Iteración 5:

En esta iteración se le dará cumplimiento a las HU #8 y #9. La cuales tienen como objetivo poder conocer las cafeterías de acuerdo a su demanda, de esta manera el centro de abastecimiento puede asignarle el producto que necesita y permitir guardar la imagen que el usuario desee, respectivamente.

3.3.3. Plan de duración de las iteraciones.

Para un mejor entendimiento y como parte del ciclo de vida de la metodología de desarrollo de software XP, se crea el plan de duración de cada una de las iteraciones que se llevarán a cabo durante el desarrollo del proyecto. Este plan tiene como finalidad mostrar la duración de cada iteración, así como el orden en que serán implementadas las HU en cada una de ellas.

Iteración	Historias de Usuario	Duración total de Iteraciones
Iteración 1	Cargar Capas.	3 semanas
	Visualizar Capas.	
	Mostrar Capas.	
Iteración 2	Desplazar Mapa.	2 semanas
Iteración 3	Aumentar Zoom.	3 semanas
	Alejar Zoom.	
	Zoom General.	
Iteración 4	Calcular Distancia.	3 semanas
Iteración 5	Analizar Demanda.	3 semanas
	Guardar Imagen.	

Tabla #13. Plan de duración de las iteraciones.

3.4. Conclusiones Parciales.

En el presente capítulo se describió la propuesta de solución para el análisis espacial de la distribución de los productos gastronómicos de la universidad, se abordaron las dos primeras fases del ciclo de vida de la metodología de desarrollo del software XP. Una de estas fases es la de exploración que a través de las historias de usuarios permitió representar una breve descripción del comportamiento del sistema y además existió una correcta comunicación entre el cliente y los desarrolladores para concretar y detallar cada historia de usuario. La fase de planificación demostró que el desarrollo del software se realizará en 5 iteraciones concluyendo que el proyecto demorará 13 semanas. También se documentaron todos los artefactos generados en el transcurso de las fases.

Capítulo 4

Construcción y validación de la solución propuesta.

4.1. Introducción.

En este capítulo se realiza una valoración de las fases de construcción y prueba de la metodología XP. Se describen las tarjetas CRC para un mejor entendimiento del sistema y se detallan las cinco iteraciones llevadas a cabo durante la etapa de construcción de la aplicación, exponiéndose las tareas de programación o ingeniería generadas por cada historia de usuario y las pruebas de aceptación efectuadas sobre el sistema.

4.2. Diseño de la solución propuesta.

La Metodología XP plantea que la implementación de un software debe realizarse de forma iterativa, obteniendo al culminar cada iteración un producto funcional que debe ser probado y mostrado al cliente para incrementar la visión de los desarrolladores con la opinión de éste. Esta es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo del software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

Esta metodología para el diseño de sus aplicaciones, no requiere la representación del sistema mediante diagramas de clase utilizando notación UML, sino que utiliza otras técnicas como las llamadas tarjetas CRC (Contenido, Responsabilidad y Colaboración). Sin embargo la utilización de estos diagramas puede aplicarse siempre y cuando influya en el mejoramiento de la comunicación entre el equipo de desarrollo y se enfoquen en la información importante. (19)

4.2.1. Tarjetas CRC.

Las tarjetas CRC permiten desprenderse del método de trabajo basado en procedimientos y trabajar con una metodología basada en objetos. Además permiten que el equipo completo contribuya en la tarea del diseño. (19)

Debido a la facilidad de su uso y entendimiento, se decidió utilizarlas para diseñar la aplicación que se desea desarrollar.

Tarjeta CRC	
Clase: Nombre de la clase que se está modelando.	
Responsabilidades: Es una descripción	Colaboraciones: Indica con cuáles

Capítulo 4: Construcción y validación

de alto nivel del propósito de la clase.	otras clases se requiere relación para cumplir la responsabilidad.
--	--

Tabla #14: Plantilla para Tarjetas CRC.

Tarjeta CRC	
Clase: ConexionPgSQL	
Responsabilidades: connec	Colaboraciones: -

Tabla #15: ConexionPgSQL.

Tarjeta CRC	
Clase: Visor	
Responsabilidades: zoomIn zoomOut pan identify select setColor zoomFull saveImag addLayer chargeLayerPostGIS chargeLayers identiDistance updateDistance loadNorth loadProgressBar closeEvent actLineEdit tematizarCafeterias getColor update add	Colaboraciones: -

Tabla #16: Visor.

Tarjeta CRC	
Clase: MapTool	
Responsabilidades: canvasReleaseEvent canvasPressEvent minDistance convertXlong convertYlat calDistance	Colaboraciones: -

Tabla #17: MapTool.

Tarjeta CRC	
Clase: Conexion	
Responsabilidades: open ejecObtConsulta getName getListCadena getListName convertToFloat getListPoints	Colaboraciones: -

Tabla #18: Conexión.

Tarjeta CRC	
Clase: GaSIG	
Responsabilidades: setupUi retranslateUi	Colaboraciones: -

Tabla #19: GaSIG.

Tarjeta CRC	
Clase: Cafeterias	

Responsabilidades: obtCafeterias points getName	Colaboraciones: -
---	--------------------------

Tabla #20: Cafeterias.

Tarjeta CRC	
Clase: Distance	
Responsabilidades: getDist getName	Colaboraciones: -

Tabla #21: Distance.

Tarjeta CRC	
Clase: Point	
Responsabilidades: getCoordX getCoordY getName	Colaboraciones: -

Tabla #22: Point.

Tarjeta CRC	
Clase: PointTema	
Responsabilidades: getVmin getVmax	Colaboraciones: -

Tabla #23: PointTema.

4.3. Desarrollo de las iteraciones.

En la segunda fase de la metodología XP, Planificación, se describieron las HU correspondiente a cada iteración a desarrollar, teniendo en cuenta las necesidades requeridas por el cliente. Durante el transcurso de las iteraciones se lleva a cabo una revisión del plan de iteraciones y se modifica en caso de ser necesario. Como parte de este plan, se descomponen las HU en tareas de programación o ingeniería, las

mismas pueden ser escritas en lenguaje técnico debido a que son para uso estricto de los programadores, donde el usuario no tiene que necesariamente comprenderlas. (19)

Teniendo en cuenta la planificación realizada se llevó a cabo el desarrollo del sistema en cinco iteraciones, obteniéndose como finalidad una aplicación con todas las condiciones y características propuestas por el cliente. A continuación se definen cada una de las iteraciones.

4.3.1. Iteración 1:

Esta iteración tiene como objetivo darle cumplimiento a las HU que se consideraron de mayor importancia para el desarrollo del software. Al concluir dicha iteración se contará con todas las funcionalidades descritas en las HU #1, #2 y #3 las cuales hacen alusión a todo lo referente con cargar, visualizar y mostrar las capas.

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Cargar Capas.	1	1
Visualizar Capas.	1	1
Mostrar Capas.	1	1

Tabla #24: HU abordadas en la primera iteración.

A continuación mediante tablas se evidencian las tareas de programación o ingeniería en las que las Historias de Usuario mencionada anteriormente fue desglosada, para un mejor funcionamiento de la aplicación.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 1
Nombre de la tarea: Cargar la cartografía del mapa a utilizar.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha inicio: 17/2/2011	Fecha fin: 24/2/2011
Programador responsable: Lisandra Lucia Sosa Nuñez.	

Capítulo 4: Construcción y validación

Descripción: Pretende cargar toda la cartografía del sistema, para luego poder trabajar en la lógica del negocio.

Tabla #25: Tarea #1 de la HU #1.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 2
Nombre de la tarea: Visualizar todas las capas de la cartografía.	
Tipo de tarea: Desarrollo.	Puntos estimados:1
Fecha inicio: 25/2/2011	Fecha fin: 5/3/2011
Programador responsable: Lisandra Lucia Sosa Nuñez.	
Descripción: Pretende visualizar todas las capas de la cartografía a trabajar en el sistema.	

Tabla #26: Tarea #1 de la HU #2

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 3
Nombre de la tarea: Mostrar todas las capas de la cartografía.	
Tipo de tarea: Desarrollo.	Puntos estimados:1
Fecha inicio: 6/3/2011	Fecha fin: 12/3/2011
Programador responsable: Lisandra Lucia Sosa Nuñez.	
Descripción: Pretende mostrar todas las capas de la cartografía a trabajar en la aplicación de acuerdo a las necesidades del usuario.	

Tabla #27: Tarea #1 de la HU #3

4.3.1. Iteración 2:

Esta iteración tiene como finalidad desarrollar la HU #4. Esta historia es la encargada de desplazar el mapa, esto le permite al usuario mover todo el lienzo o área de trabajo según le sea necesario.

	Tiempo de Implementación (semanas)
--	------------------------------------

Capítulo 4: Construcción y validación

Historias de Usuario	Estimación	Real
Desplazar Mapa.	2	2

Tabla #28: HU abordadas en la segunda iteración.

A continuación mediante tablas se evidencian las tareas de programación o ingeniería en las que la Historia de Usuario mencionada anteriormente fue desglosada, para un mejor funcionamiento de la aplicación.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 4
Nombre de la tarea: Desplazar el mapa hacia cualquier punto del lienzo.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Fecha inicio: 12/3/2011	Fecha fin: 26/3/2011
Programador responsable: Lisandra Lucia Sosa Nuñez.	
Descripción: Consiste en lograr una mejor visibilidad sobre cualquier punto que el usuario referencie en el lienzo.	

Tabla #29: Tarea #1 de la HU #4.

4.3.2. Iteración 3:

En esta iteración se dará cumplimiento a las HU que corresponden a los números #5, #6 y #7. Estas son referentes al zoom del mapa, le permiten al cliente acercar y alejar la parte seleccionada del mapa, así como su vista completa.

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Aumentar Zoom.	1	1
Alejar Zoom.	1	1
Zoom General.	1	1

Tabla #30: HU abordadas en la tercera iteración.

Capítulo 4: Construcción y validación

A continuación mediante tablas se evidencian las tareas de programación o ingeniería en las que las Historias de Usuario mencionada anteriormente fue desglosada, para un mejor funcionamiento de la aplicación.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 5
Nombre de la tarea: Aumentar referencia del mapa.	
Tipo de tarea: Desarrollo.	Puntos estimados:1
Fecha inicio: 27/3/2011	Fecha fin: 2/4/2011
Programador responsable: Lisandra Lucia Sosa Nuñez.	
Descripción: Pretende lograr una mayor visibilidad de los detalles del mapa.	

Tabla #31: Tarea #1 de la HU #5.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 6
Nombre de la tarea: Disminuir referencia del mapa.	
Tipo de tarea: Desarrollo.	Puntos estimados:1
Fecha inicio: 9/4/2011	Fecha fin: 16/4/2011
Programador responsable: Lisandra Lucia Sosa Nuñez.	
Descripción: Pretende quitar visibilidad de los detalles del mapa.	

Tabla #32: Tarea #1 de la HU #6.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 7
Nombre de la tarea: Trasladar el mapa a escala normal.	
Tipo de tarea: Desarrollo.	Puntos estimados:1
Fecha inicio: 17/4/2011	Fecha fin:23/4/2011

Programador responsable: Lisandra Lucia Sosa Nuñez.
Descripción: Consiste en volver a mostrar el mapa en su escala normal, para que el usuario pueda tomar otro tipo de referencia.

Tabla #33: Tarea #1 de la HU #7.

4.3.3. Iteración 4:

Esta iteración tiene como propósito desarrollar la HU #8. Esta historia es la que se encarga de calcular la distancia de las cafeterías.

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Calcular Distancia.	3	3

Tabla #34: HU abordadas en la cuarta iteración.

A continuación mediante tablas se evidencian las tareas de programación o ingeniería en las que la HU mencionada anteriormente fue desglosada, para un mejor funcionamiento de la aplicación.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 8
Nombre de la tarea: Conocer las distancias de todas las cafeterías.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Fecha inicio: 24/4/2011	Fecha fin: 7/5/2011
Programador responsable: Lisandra Lucia Sosa Nuñez.	
Descripción: Pretende calcular la distancia desde el punto de origen hacia todas las cafeterías.	

Tabla #35: Tarea #1 de la HU #8.

Debido a que el QGis no deja acceder a la clase MeasureLine por encontrarse en el paquete de clases de QGis app, fue necesario implementar las funcionalidad calcular distancia. Desarrollando esta en forma de radio, es decir desde el punto x,y del ratón (mouse) seleccionado en el mapa calcular la distancia en forma de radio hacia todos los puntos de referencias donde se encuentra ubicadas las cafeterías.

Pasos para calcular todas las distancias desde el punto de origen hacia todas las cafeterías.

1. Seleccionar el punto cuando el usuario da clic.
2. Tomar las coordenadas de x;y.
3. Adicionar a una lista de puntos las coordenadas x;y de cada cafetería en coordenadas geográficas.
4. Se transforma el x;y del mouse de pixeles a coordenadas geográficas (longitud, latitud):

longitud = $\text{pix2Geo}(x, \text{valor mínimo de pixel para la } x, \text{ valor máximo de pixel para } x, \text{ valor mini } x \text{ en el extensión del mapa, valor máximo de } x \text{ en la extensión del mapa})$

latitud = $\text{pix2Geo}(y, \text{valor mínimo de pixel para la } y, \text{ valor máximo de pixel para } y, \text{ valor mini } y \text{ en el extensión del mapa, valor máximo de } y \text{ en la extensión del mapa})$

Cálculo: $\text{geoMin} + (\text{pixPos} - \text{pixMin}) * (\text{geoMax} - \text{geoMin}) / (\text{pixMax} - \text{pixMin})$

5. Se calcula la distancia entre dos puntos por la fórmula cartesiana:

punto1:(punto1.x, punto1.y)

punto2:(punto2.x, punto2.y)

Cálculo: $\text{Math.sqrt}(\text{Math.pow}((\text{punto2.x}-\text{punto1.x}), 2)+\text{Math.pow}((\text{punto2.y}-\text{punto1.y}), 2))$

6. Se convierte a metro utilizando el radio de la tierra:

Radio (ecuatorial) de la tierra en metros = 6378400

punto1:(punto1.x, punto1.y)

punto2:(punto2.x, punto2.y)

Cálculo de diferencia de latitud y longitud:

longitud1 = punto1.x * Math.PI/180

Capítulo 4: Construcción y validación

latitud1 = punto1.y * Math.PI/180

longitud2 = punto2.x * Math.PI/180

latitud2 = punto2.y * Math.PI/180

longitud = (longitud2 - longitud1)

latitud = (lat2 - lat1)

Cálculos intermedios:

$a = \text{Math.pow}((\text{Math.sin}(\text{latitud}/2)),2) + \text{Math.cos}(\text{latitud1}) * \text{Math.cos}(\text{latitud2}) * \text{Math.pow}(\text{Math.sin}(\text{longitud}/2),2)$

$b = 2 * \text{Math.atan2}(\text{Math.sqrt}(a), \text{Math.sqrt}(1-a))$

Cálculo Final: Radio * b

Tareas de la Ingeniería	
No. de la tarea: 2	No. de la HU: 8
Nombre de la tarea: Conocer la distancia de la cafetería más cercana.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha inicio: 8/5/2011	Fecha fin: 14/5/2011
Programador responsable: Lisandra Lucia Sosa Nuñez.	
Descripción: Pretende devolver la cafetería más cercana al punto de origen.	

Tabla #36: Tarea #2 de la HU #8.

4.3.4. Iteración 5:

En esta iteración se le dará cumplimiento a las HU #9 y #10. Las cuales tienen como propósito conocer las cafeterías de acuerdo a su demanda y guardar una imagen del mapa.

Historias de Usuario	Tiempo de Implementación (semanas)	
	Estimación	Real
Analizar Demanda.	2	2

Capítulo 4: Construcción y validación

Guardar Imagen.	1	1
-----------------	---	---

Tabla #37: HU abordadas en la quinta iteración.

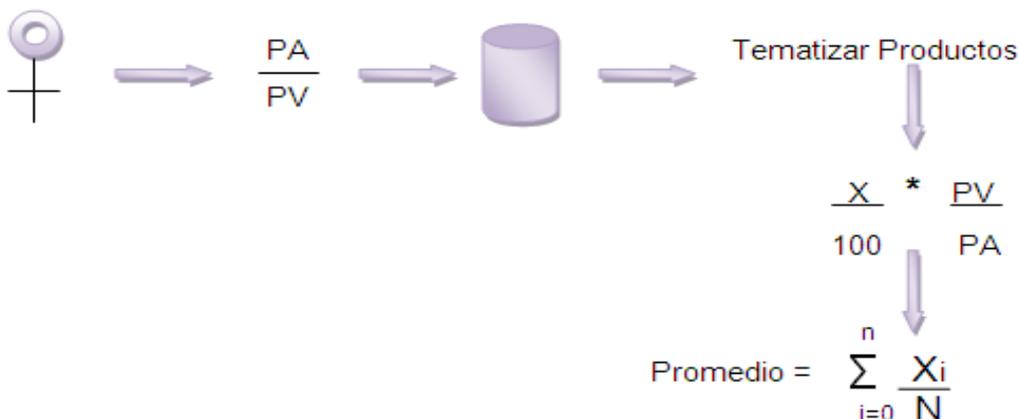
A continuación mediante tablas se evidencian las tareas de programación o ingeniería en las que las Historias de Usuario mencionadas anteriormente fueron desglosadas, para un mejor funcionamiento de la aplicación.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 9
Nombre de la tarea: Demanda de productos por cafeterías.	
Tipo de tarea: Desarrollo.	Puntos estimados: 2
Fecha inicio: 15/5/2011	Fecha fin: 28/5/2011
Programador responsable: Lisandra Lucia Sosa Nuñez.	
Descripción: Esta tarea consiste en colorear la cafetería de acuerdo a su demanda.	

Tabla #38: Tarea #1 de la HU #9.

Para realizar la tarea de ingeniería que lleva como nombre demanda de productos por cafeterías se utilizó la funcionalidad `GraduatedSymbolRender` de la clase `QgsGraduatedSymbolRenderer` que se encuentra en el paquete de clases de `QGis core`. Esta brinda la posibilidad de pasar el nombre de la capa siendo capaz de coger las columnas comparables que existen en la tabla `puntosventas` ubicada en la base de datos `SIGUCITesis` de forma tal que se especifica el campo a comparar, devolviendo este el polígono y dándole la posibilidad de asignarle un color determinado. Además esta clase trae como ventaja que los rangos de comparación ya vienen definidos.

Pasos para conocer las cafeterías de acuerdo a su demanda.



X: % de demanda.

N: Total de Xi.

1. Se insertan en la base de datos SIGUCITesis la cantidad de productos asignados y vendidos de cada cafetería.
2. Teniendo los campos asignados y vendidos de la tabla puntosventas de la base de datos SIGUCITesis, se realiza el cálculo para obtener el valor estadístico.

Cálculo:

% = vendidos * 100 / asignados.

$$\text{promedio} = \frac{\sum_{i=0}^n X_i}{N}$$

3. Se actualiza la base de datos SIGUCITesis insertando el valor estadístico.
4. De acuerdo a ese valor estadístico (Demanda por producto) se colorean las cafeterías.

Tareas de la Ingeniería	
No. de la tarea: 1	No. de la HU: 10
Nombre de la tarea: Guardar imagen.	
Tipo de tarea: Desarrollo.	Puntos estimados: 1
Fecha inicio: 29/5/2011	Fecha fin: 4/6/2011
Programador responsable: Lisandra Lucia Sosa Nuñez.	
Descripción: Consiste en guardar la imagen que el usuario desee y en la escala que este tenga.	

Tabla #39: Tarea #1 de la HU #10.

4.4. Requisitos no funcionales del sistema.

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Estas propiedades son las características que hacen al producto atractivo, usable, rápido y confiable.

Usabilidad

RNF 1. El sistema está creado para ser usado por personas que tengan conocimientos básicos de informática y de los Sistemas de Información Geográfica.

Fiabilidad

RNF 2. El sistema debe estar disponible todo el tiempo para sus usuarios, descontando el tiempo en que se encuentre en mantenimiento.

RNF 3. El período entre fallos recuperables, como por ejemplo fallos en el servidor no debe exceder las 24 horas.

Soporte

RNF 4. Cualquier cambio ocurrido en la aplicación ya sea agregar otras tablas o otros campos a la base de datos SIGUCITesis, es deber del líder de proyecto asegurarse que se realice la actualización en el servidor.

Restricciones de diseño

RNF 5. El producto de software final debe diseñarse sobre una arquitectura modelo-vista-controlador.

RNF 6. Se deben utilizar los estándares determinados (diseño de interfaces, base de datos y codificación).

Interfaz

Interfaces de usuario

El sistema debe:

RNF 7. El sistema debe mostrar de forma organizada las funcionalidades del sistema.

RNF 8. El sistema debe tener un diseño sencillo, donde no sea necesario mucho entrenamiento para ser utilizado.

RNF 9. Los botones utilizados en la aplicación deben tener un tamaño modelado y su nombre entendible para el usuario.

Requisitos de hardware

Para los servidores:

RNF 10. Se requiere tarjeta de red.

RNF 11. El servidor de base de datos tenga como mínimo 1GB de RAM y 40GB de disco duro o más.

Interfaces de software

Para los Servidores:

RNF 12. Sistemas operativos Ubuntu versión 9.4 en adelante.

RNF 13. PostgreSQL como Sistema Gestor de Base de Datos.

RNF 14. PostGIS como extensión de PostgreSQL como soporte de datos espaciales.

RNF 15. Python versiones 2.5 en adelante.

RNF 16. Eclipse como entorno integrado de desarrollo del software.

RNF 17. QGIS versión 1.1 en adelante.

RNF 18. Las bindings¹¹ de Python para QGIS (es decir, PyQGIS¹²).

RNF 19. Las bindings de Python para Qt4 (es decir, PyQt4¹³).

RNF 20. QtSQL para el acceso a PostgreSQL.

Requisitos de Licencia

RNF 21. La arquitectura de acuerdo a los tipos de licencias de las herramientas que se van a utilizar es legalmente de modelo libre, permitiendo la utilización, modificación y distribución de las mismas por terceros sin necesidad de obtener la autorización de sus respectivos titulares.

Requisitos Legales, de Derecho de Autor y otros

RNF 22. La mayoría de las herramientas de desarrollo son libres y del resto, las licencias están avaladas.

Estándares Aplicables

RNF 18. El sistema será desarrollado bajo estándares internacionales como la normativa ISO19115.

4.5. Pruebas.

Una de las prácticas de la metodología XP es el uso de las pruebas para garantizar el funcionamiento de los códigos que se vayan implementando. Esto permite aumentar la calidad de los sistemas reduciendo el número de errores no detectados y disminuyendo el tiempo transcurrido entre la aparición de un error y su detección. También permite aumentar la seguridad de evitar efectos colaterales no deseados a la hora de realizar modificaciones y refactorizaciones.

La metodología ágil XP divide las pruebas en dos grupos: pruebas unitarias y pruebas de aceptación. Las pruebas unitarias son desarrolladas por los programadores y se encargan de verificar el código automáticamente y las pruebas de aceptación están destinadas a verificar que al final de cada iteración las Historias de Usuario cumplen con la funcionalidad asignada y satisfagan las necesidades del cliente. (28)

¹¹ Bindings: Enlaces.

¹² PyQGIS: Permite utilizar funciones y clases que se encuentran en el núcleo de QGIS como: las bibliotecas core y ui para ser utilizadas en Python.

¹³ PyQt4: Permite utilizar todas las bibliotecas de Qt4 en Python para crear interfaces gráfica.

4.5.1. Pruebas unitarias.

De acuerdo con lo que plantea la metodología XP, las pruebas unitarias o pruebas de unidad consisten en comprobaciones (manuales o automatizadas) desarrolladas por los programadores. Las cuales se realizan para verificar que el código correspondiente a un módulo concreto se comporta de manera esperada.

Las pruebas unitarias proporcionan beneficios tales como: (28)

- Brindan al programador una inmediata retroalimentación de como está realizando su trabajo.
- El programador puede realizar cambios de forma segura respaldado por efectivos casos de prueba.
- Permite saber si una determinada funcionalidad se puede agregar al sistema existente sin alterar el funcionamiento actual del mismo.

Las pruebas unitarias no se le aplicarán al software debido a que según los expertos en la metodología de desarrollo XP recomiendan utilizar las pruebas de aceptación. Las pruebas de aceptación son consideradas las más adecuadas, pues significa el grado de satisfacción que tenga el cliente con el producto. En estas pruebas el cliente puede comprobar si todas las historias de usuario se implementaron de acuerdo a lo concebido.

4.5. Pruebas de aceptación.

Las pruebas de aceptación son más importantes que las pruebas unitarias dado que significan la satisfacción del cliente con el producto desarrollado y el final de una iteración y el comienzo de la siguiente. Las pruebas de aceptación se elaboran a lo largo de la iteración, en paralelo con el desarrollo del sistema, y adaptándose a los cambios que el sistema sufra.

Las pruebas de aceptación son consideradas como “pruebas de caja negra”. Los clientes son responsables de verificar que los resultados de estas pruebas sean correctos. Así mismo, en caso de que fallen varias pruebas, deben indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación. (28)

Caso de prueba de aceptación

Código: HU3_PA1

No. de la HU: 3

Nombre: Comprobar que las capas del mapa se muestren.

Capítulo 4: Construcción y validación

Descripción: Evaluar que las capas del mapa se muestren de acuerdo a las necesidades del usuario.
Condiciones de Ejecución: El cliente debe probar que las capas del mapa se muestren correctamente.
Entrada/ Pasos de ejecución: Se solicita la opción capas para probar que el cliente pueda elegir que capa mostrar.
Resultado Esperado: Se mostraron correctamente las capas que el cliente eligió.
Evaluación de la Prueba: Prueba satisfactoria.

Tabla #40: Prueba de aceptación para la HU "Mostrar Mapa."

Caso de prueba de aceptación	
Código: HU4_PA1	No. de la HU: 4
Nombre: Comprobar el desplazamiento del mapa.	
Descripción: Evaluar el desplazamiento del mapa sobre el lienzo.	
Condiciones de Ejecución: El cliente debe probar que se desplace correctamente el mapa sobre el lienzo.	
Entrada/ Pasos de ejecución: Se solicita la opción panear para probar el desplazamiento del mapa.	
Resultado Esperado: Se produce correctamente el desplazamiento del mapa.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla #41: Prueba de aceptación para la HU "Desplazar Mapa."

Caso de prueba de aceptación	
Código: HU5_PA1	No. de la HU: 5
Nombre: Aumentar Zoom.	
Descripción: Evaluar que la imagen seleccionada aumente de tamaño.	
Condiciones de Ejecución: El cliente debe probar que la imagen aumente correctamente.	
Entrada/ Pasos de ejecución: Se solicita la opción aumentar para poder probar que la imagen aumente.	

Capítulo 4: Construcción y validación

Resultado Esperado: Se aumenta correctamente la imagen.

Evaluación de la Prueba: Prueba satisfactoria.

Tabla #42: Prueba de aceptación para la HU “Aumentar Zoom.”

Caso de prueba de aceptación	
Código: HU6_PA1	No. de la HU: 6
Nombre: Alejar Zoom.	
Descripción: Evaluar que la imagen seleccionada se aleje.	
Condiciones de Ejecución: El cliente debe probar que la imagen se aleje correctamente.	
Entrada/ Pasos de ejecución: Se solicita la opción alejar para poder probar que la imagen se aleja.	
Resultado Esperado: Se aleja correctamente la imagen.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla #43: Prueba de aceptación para la HU “Alejar Zoom.”

Caso de prueba de aceptación	
Código: HU7_PA1	No. de la HU: 7
Nombre: Zoom General.	
Descripción: Evaluar que la imagen vuelva a su proporción original.	
Condiciones de Ejecución: El cliente debe probar que la imagen vuelva correctamente a su proporción original.	
Entrada/ Pasos de ejecución: Se solicita la opción vista completa para poder probar que la imagen vuelve a su proporción original.	
Resultado Esperado: Se genera correctamente que la imagen vuelva a su proporción original	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla #44: Prueba de aceptación para la HU “Zoom General.”

Capítulo 4: Construcción y validación

Caso de prueba de aceptación	
Código: HU8_PA1	No. de la HU: 8
Nombre: Evaluar el cálculo de la distancia.	
Descripción: Comprobar que dado un punto de origen se obtenga la distancia de la cafetería mas cercana.	
Condiciones de Ejecución: El cliente debe probar que dado un punto de origen se obtenga correctamente la distancia de la cafetería mas cercana	
Entrada/ Pasos de ejecución: Se solicita la opción calcular distancia y se da un click sobre el lienzo para poder probar que devuelva la distancia de la cafetería más cercana.	
Resultado Esperado: Se obtiene la cafetería más cercana al punto de origen.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla #45: Prueba de aceptación para la HU “Calcular Distancia.”

Caso de prueba de aceptación	
Código: HU9_PA1	No. de la HU: 9
Nombre: Evaluar la tematización por demanda.	
Descripción: Comprobar que dado el producto del que se desea conocer su demanda, se colorean correctamente las cafeterías que se encuentran de acuerdo al rango.	
Condiciones de Ejecución: El cliente debe probar que dado el producto del que se desea conocer su demanda, se colorean correctamente las cafeterías que se encuentran de acuerdo al rango.	
Entrada/ Pasos de ejecución: Primero se solicita la opción analizar demanda, luego se selecciona el producto del que se desea conocer su demanda.	
Resultado Esperado: Se obtiene la tematización de las cafeterías por demanda.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla #46: Prueba de aceptación para la HU “Analizar Demanda.”

Caso de prueba de aceptación	
Código: HU10_PA1	No. de la HU: 10
Nombre: Comprobar que la imagen se guarde.	
Descripción: Evaluar que la imagen se guarde con la escala que se desee.	
Condiciones de Ejecución: El cliente debe probar que la imagen se guarde correctamente con la escala que escogió.	
Entrada/ Pasos de ejecución: Se solicita la opción guardar imagen para poder probar que la imagen se guarda con la escala deseada.	
Resultado Esperado: Se guarda correctamente la imagen con escala deseada.	
Evaluación de la Prueba: Prueba satisfactoria.	

Tabla #47: Prueba de aceptación para la HU "Guardar Imagen."

Se realizó una prueba piloto donde se tomó una muestra de 10 personas para ver el correcto funcionamiento del sistema. Los parámetros a seguir fueron:

- 1- Interfaz amigable.
- 2- Funcionalidad aceptar.
- 3- Funcionalidad alejar.
- 4- Funcionalidad vista completa.
- 5- Funcionalidad desplazar.
- 6- Funcionalidad distancia.
- 7- Funcionalidad demanda.
- 8- Funcionalidad guardar imagen.
- 9- Utilidad de la herramienta para el análisis espacial de la distribución de los productos.

Resultado de las pruebas de aceptación.

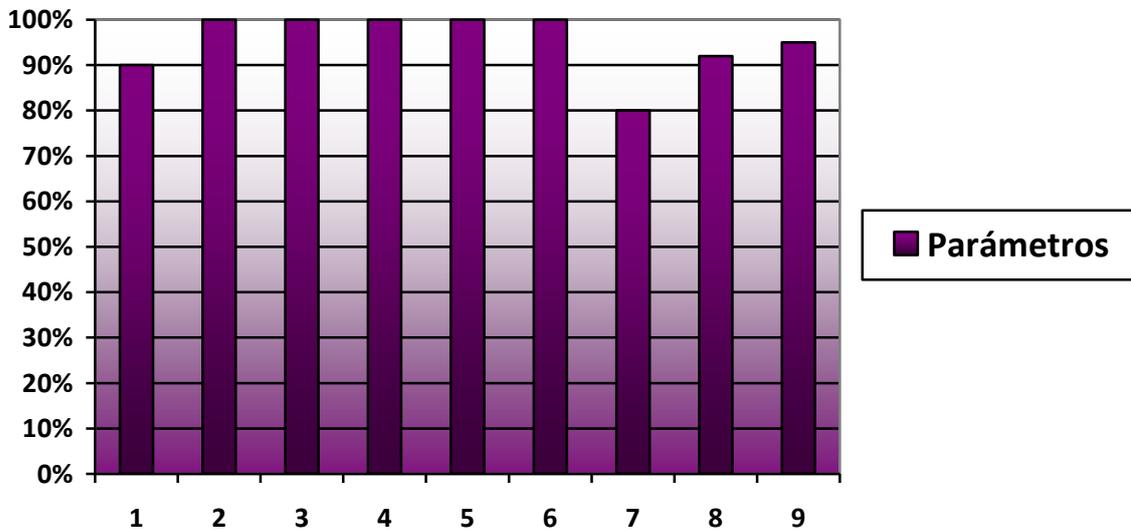


Tabla #48: Resultados de las pruebas de aceptación.

4.6. Conclusiones Parciales.

En este capítulo se realizó la construcción y validación de la solución propuesta para el desarrollo de la aplicación, donde se abordaron las fases de construcción y prueba de la metodología de desarrollo del software XP. Se describieron las tarjetas CRC para lograr un mejor entendimiento del sistema. Se detallaron las cinco iteraciones llevadas a cabo durante la etapa de construcción de la aplicación, así como las tareas de programación o ingeniería generadas por cada historia de usuario. Se realizó un profundo estudio sobre las pruebas unitarias y de aceptación, seleccionando estas últimas como las más indicadas para comprobar el funcionamiento del software, debido a que demuestran la satisfacción del cliente con el producto.

Conclusiones Generales

Con el desarrollo del Sistema de Información Geográfica para el análisis espacial de la distribución de los productos gastronómicos de la Universidad de las Ciencias Informáticas y luego de la realización de las tareas de investigación científicas, se ha arribado a las siguientes conclusiones.

- El estudio realizado para la investigación permitió conocer que no existía una correcta distribución de los alimentos gastronómicos en los diferentes lugares de servicios de la universidad.
- Las herramientas, la metodología de desarrollo del software y el lenguaje utilizado, brindaron el soporte necesario para lograr un producto con los requerimientos deseados, además de proporcionarle al mismo una calidad y un rendimiento acordes a las exigencias planteadas por el cliente.
- Se analizó el funcionamiento de la aplicación, y la misma respondió satisfactoriamente todas las pruebas.
- El Sistema de Información Geográfica se encuentra listo para ser utilizado por los directivos gastronómicos.

Se ha cumplido el objetivo de la investigación, pues se obtuvo un Sistema de Información Geográfica que permite apoyar el mejoramiento en el área de los servicios gastronómicos de la universidad.

Recomendaciones

Las metas planteadas con este trabajo se han cumplido y en base a los resultados obtenidos se recomienda:

- Agregar un módulo que permita la edición cartográfica para facilitar a los directivos de gastronomía el trabajo en el proceso de distribución de los alimentos.
- Cuando esté disponible el grafo de la cartografía de viales de la universidad, incorporar la funcionalidad del camino mínimo utilizando Pg-Routing embebida en el sistema de gestor de base dato Postgres 8.4.
- Desarrollo de un sistema web que permita la actualización de los datos asociados a las cafeterías por un especialista.
- Un modelo matemático que represente con mayor precisión el valor estadístico relacionado con la demanda de los productos y el costo de los mismos.

Referencias Bibliográficas

1. Sistema de información geográfica en línea. [En línea] [Citado el: 24 de 11 de 2010.] http://sigeco.ecologia.campeche.gob.mx/info_sig.php.
2. Pumain, Denise. HyperGeo. [En línea] [Citado el: 24 de 11 de 2010.] http://www.hypergeo.eu/article.php3?id_article=265.
3. Kafati, Elizabeth Gutierrez. Los sistemas de información Geográfica SIG. [En línea] 30 de 4 de 2009. [Citado el: 25 de 11 de 2010.] <http://manuelgross.bligoo.com/content/view/501371/Los-sistemas-de-informacion-Geografica-SIG.html>.
4. Dr. José Luis Batista Silva Investigador Titular, Instituto de Geografía Tropical Especialista Ramal,. APLICACIÓN DE SISTEMAS DE INFORMACIÓN GEOGRÁFICA EN CUBA. [En línea] http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1051.
5. El software geográfico para la gestion del territorio. [En línea] [Citado el: 26 de 11 de 2010.] <http://www.softwaregis.cl/arcgis.html>.
6. Procalculo Prosis. [En línea] [Citado el: 26 de 11 de 2010.] <http://www.procalculoopsis.com/Productos/ESRI-ArcGIS-Desktop/ArcInfo.aspx>.
7. *Manual de Quantum Gis*.
8. Los sistemas de información Geográfica SIG. [En línea] <http://manuelgross.bligoo.com/content/view/501371/Los-sistemas-de-informacion-Geografica-SIG.html..>
9. Duque, Raul Gonzalez. *Python para todos*. España : s.n.
10. *Manual de Eclipse*
11. Garrido, Salvador Alemany. *Introducción a Qt*. 2009.
12. Pixelco blog. Un blog sobre diseño y desarrollo web, Internet y Tecnología. [En línea] 2011. <http://pixelcoblog.com/qt-creator-completo-entorno-de-desarrollo-multiplataforma/>.
13. Lockhart, Thomas. *Manual del usuario de PostgreSQL*. California : s.n., 1996.

14. [En línea] http://www.postgresql-es.org/sobre_postgresql.
15. Arquba.com.ar. [En línea] <http://www.arquba.com.ar/software-gratis/quantum-gis/>.
16. González, Isaías Carrillo Pérez y Rodrigo Pérez. METODOLOGIA DE DESARROLLO DEL SOFTWARE. 2008.
17. Solis, Figueroa. "Metodologías Tradicionales vs. Metodologías Ágiles." . 2010.
18. Ogrissek, Lehmann y. *Los sistemas de información geográfica y la telepercepción en la pesca*. 1988.
19. Joskowicz, Ing. José. *Reglas y Prácticas en extreme Programming*. 2008.
20. Duffatt, Luís R. Díaz Cisneros y Rafael Candeaux. *Revista Internacional de Ciencias de la Tierra*. 1994.
21. Tohmé, Juan M.C. Larrosa - Fernando. *Redes de equilibrio con comunicación bidireccional: Redes lineales con nodos de activación intermedios*.
22. Bijit, Leopoldo Silva. Capítulo 6: Redes lineales. 27-06-2008.
23. Nieto, José Heber. *Notas sobre Redes de Flujo*.
24. Universidad Técnica Particular de Loja. *Matemática Discreta 2010*. [En línea] 2010. [Citado el 25 de 1 de 2011.] <http://blogs.utpl.edu.ec/matematicadiscreta2010/2010/05/27/modelo-de-redes/>.
25. Grass. [En línea] 2010. [Citado el: 12 de 12 de 2010.] www.scribd.com/doc/2096954/SIG-Libres..
26. Olamendi, Gabriel. *Distribución*. 2009.
27. Basterra, Ing. Indiana. .(FACULTAD DE INGENIERÍA CÁTEDRA DE FOTOINTERPRETACIÓN SISTEMAS DE INFORMACIÓN GEOGRÁFICA. [En línea] [Citado el: 14 de 12 de 2010.]
28. J. J. Gutiérrez, M. J. Escalona, M. Mejías, J. Torres. *PRUEBAS DEL SISTEMA EN PROGRAMACIÓN EXTREMA*. España : s.n.

Bibliografía Consultada

1. Kafati, Elizabeth Gutierrez. Los sistemas de información Geográfica SIG. [En línea] 30 de 4 de 2009. [Citado el: 25 de 11 de 2010.]
2. <http://manuelgross.bligoo.com/content/view/501371/Los-sistemas-de-informacion-Geografica-SIG.html>.
3. Dr. José Luis Batista Silva Investigador Titular, Instituto de Geografía Tropical Especialista Ramal,. APLICACIÓN DE SISTEMAS DE INFORMACIÓN GEOGRÁFICA EN CUBA. [En línea] http://www.mappinginteractivo.com/plantilla-ante.asp?id_articulo=1051.
4. Procalculo Prosis. [En línea] [Citado el: 26 de 11 de 2010.] <http://www.procalculoproxis.com/Productos/ESRI-ArcGIS-Desktop/ArcInfo.aspx>.
5. *Manual de Quantum Gis*.
6. Duque, Raul Gonzalez. *Python para todos*. España : s.n.
7. *Manual de Eclipse*
8. Garrido, Salvador Alemany. *Introducción a Qt*. 2009.
9. González, Isaías Carrillo Pérez y Rodrigo Pérez. METODOLOGIA DE DESARROLLO DEL SOFTWARE. 2008.
10. Solis, Figueroa. "*Metodologías Tradicionales vs. Metodologías Ágiles*." . 2010.
11. Joskowicz, Ing. José. *Reglas y Prácticas en extreme Programming*. 2008.
12. Tohmé, Juan M.C. Larrosa - Fernando. *Redes de equilibrio con comunicación bidireccional:Redes lineales con nodos de activación intermedios*.
13. Universidad Técnica Particular de Loja. *Matemática Discreta 2010*. [En línea] 2010. [Citado el 25 de 1 de 2011.] <http://blogs.utpl.edu.ec/matematicadiscreta2010/2010/05/27/modelo-de-redes/>.
14. Nieto, José Heber. *Notas sobre Redes de Flujo*.
15. J. J. Gutiérrez, M. J. Escalona, M. Mejías, J. Torres. *PRUEBAS DEL SISTEMA EN PROGRAMACIÓN EXTREMA*. Espana : s.n.

Bibliografía Consultada

16. Lockhart, Thomas. *Manual del usuario de PostgreSQL*. California : s.n., 1996.

17. [En línea] http://www.postgresql-es.org/sobre_postgresql.